計算機数学 I 講義ノート

数理物質科学研究科 数学専攻 坂井公

(平成 29 年 4 月 6 日)

目 次

1 準備 **8**
1.1 コンピュータの基本構成 8
1.2 漸近記法 10

2 数値と数式の表現，基本演算アルゴリズム **12**
2.1 数値の表現 12
2.1.1 整数の表記（2 の補数） 12
2.1.2 実数の表記（浮動小数点数） 13
2.1.3 自然数の表記（グレイコード）★ 14
2.1.4 整数の表記（−2 進）★ 14
2.1.5 多精度整数の表記と加算 15
2.2 多項式の表現 17
2.2.1 多項式の表現と加算 17
2.2.2 ホーナー法 18
2.2.3 非負整数の 2 進 10 進変換 19
2.2.4 小数・分数の 2 進 10 進変換 20
2.3 乗算 21
2.3.1 多項式の乗算 21
2.3.2 多精度整数と単精度整数の乗算 22
2.3.3 多精度整数と多精度整数の乗算 23
2.4 剰余つき除算 23
2.4.1 多項式の剰余つき除算 24
2.4.2 多精度整数の剰余つき除算 25

3 整数の基本的性質 ☆ **26**
3.1 整除性 ☆ 26
3.1.1 整除性と素数 ☆ 26
3.1.2 最大公約数 ☆ 28
3.1.3 素因数分解の一意性 ☆ 28

3.2 合同関係 ☆ 32
3.2.1 合同と法計算 ☆ 32
3.2.2 線形合同式 ☆ 34
3.2.3 中国剰余定理 ☆ 35
3.3 剰余類 36
3.3.1 剰余類 36
3.3.2 オイラーの φ 関数 ☆ 37
3.3.3 オイラーの定理とフェルマーの小定理 ☆ 38
3.4 平方剰余 39
3.4.1 奇素数を法とする平方剰余 40
3.4.2 奇素数のベキを法とする平方剰余 41
3.4.3 2^e を法とする平方剰余 42
3.4.4 一般の正整数を法とする平方剰余 42
3.4.5 奇素数を法とする -1 の平方根 43
3.5 ディリクレ積とメビウス関数★ 45

4 ユークリッド互除法（The Euclidean Algorithm） 49
4.1 イデアル，剰余環，GCD，LCM ☆ 49
4.2 ユークリッド整域と互除法 51
4.3 拡張ユークリッド互除法 53
4.4 拡張ユークリッド互除法の計算量 56
4.5 ユークリッド関数の性質について★ 57
4.6 GCD の一意性について★ 58
4.7 法逆元計算（Modular inverse） 60
4.8 中国剰余算法（Chinese remainder algorithm） 63
4.9 行列積の法計算 (Modular Matrix Multiplication) 64
4.10 行列式の法計算 (Modular Determinant Computation) 66
4.11 フェルマの 2 平方定理—計算版 67
4.12 分数小数変換 70
4.13 連分数 (continued fraction) と実数の有理数近似 74
4.13.1 カレンダー (Calendars) ★ 76
4.13.2 音階 (Musical scales) ★ 77

5 反覆法 80
5.1 縮小写像の原理 80
5.2 Newton 法 81
5.3 加速法 82
5.4 Steffensen 反復 83

6 FFT と高速乗算法 **85**
6.1 多項式の関数値表現 85
6.1.1 秘密共有 (Secret Sharing) 87
6.2 高速フーリエ変換 87
6.2.1 1 の複素 n 乗根と離散フーリエ変換 87
6.2.2 高速フーリエ変換 88
6.2.3 逆変換と多項式の高速乗算法 89
6.3 整数係数多項式の FFT による積と法計算 90

記法など

数やその集合

$\mathbb{N}$	非負の整数の全体（0 も含まれることに注意）
$\mathbb{Z}$	整数の全体
$\mathbb{Q}$	有理数の全体
$\mathbb{R}$	実数の全体
$\mathbb{C}$	複素数の全体
i	虚数単位 $\sqrt{-1}$
$[a..b]$	閉区間 $\{x \mid a \leqq x \leqq b\}$ (他と混用の多い $[a, b]$ は避ける)
$(a..b)$	開区間 $\{x \mid a < x < b\}$ (他と混用の多い (a, b) は避ける)
$(a..b]$	半開区間 $\{x \mid a < x \leqq b\}$
$[a..b)$	半開区間 $\{x \mid a \leqq x < b\}$
$\lfloor x \rfloor$	x を超えない最大の整数 (混用の多いガウスの記号 $[x]$ は避ける)
$\lceil x \rceil$	x 以上の最小の整数
$\Re(z)$，$\Im(z)$	複素数 z の実数部と虚数部 (すなわち $z = \Re(z) + \Im(z)\mathrm{i}$ である)
z^*	複素数 z の共役 $\Re(z) - \Im(z)\mathrm{i}$
	（他書では，$\bar{z}$ であらわすことも多いが，本ノートでは随伴行列と同じ記法を用いる。）
$\lvert z \rvert$	複素数 z の絶対値 $\sqrt{zz^*}$
$\exp(x)$	指数関数 e^x
$\log x$	常用対数 $\log_{10} x$
$\ln x$	自然対数 $\log_e x$
$\lg x$	底を 2 とする対数 $\log_2 x$

集合や関数

$\mathbf{On}$	順序数の全体
$\#X$	集合 X の要素数
Y^X	集合 X から集合 Y への関数の全体
2^X	(上の記号の流用) 集合 X の部分集合の全体
$\mathbf{id}_X$	X から X への恒等関数
$\mathrm{S}(X)$	X 上の置換（X から X への全単射）の全体
$f \circ g$	関数 f と g の合成。すなわち $(f \circ g)x = f(g(x))$
S_n	n 次対称群。すなわち $\{1, \ldots, n\}$ 上の置換の全体 $\mathrm{S}\{1, \ldots, n\}$
A_n	n 次交代群。すなわち $\{1, \ldots, n\}$ 上の偶置換の全体
$(i_1, \cdots, i_k)$	巡回置換。特に (i, j) は i と j を入れ替える互換
$\mathbf{id}_n$	$\{1, \ldots, n\}$ 上の恒等置換 $\mathbf{id}_{\{1,\ldots,n\}}$
$\operatorname{sgn}\sigma$	置換 σ の符号（奇置換なら -1，偶置換なら 1）
$[P]$	Iverson の記法（下記参照）
$K[x]$	変数 x についての K 係数多項式の全体（K は一般には環）
$K[x]_n$	変数 x についての n 次以下の K 係数多項式の全体（K は一般には環）

P を任意の命題とするとき，Iverson の記号 $[P]$ は，P が真なら $[P] \stackrel{\text{def}}{=} 1$，$P$ が偽なら $[P] \stackrel{\text{def}}{=} 0$ と定義される。例えば，$[1 < 2] = 1$，[4 は素数] $= 0$ である。また，クロネッカーの δ は，$\delta_{ij} = [i = j]$ と書ける。

行列とベクトルやその成分

$\vec{x}, \vec{y}$ など	（普通は縦）ベクトル
$\vec{e}_i$	基本ベクトル（一般には基底ベクトル）
$\vec{0}$	ゼロベクトル
$\mathrm{M}(m \times n, K)$	体 K の要素を成分とする m 行 n 列の行列全体。$K^{m\times n}$ と記すことも多い。（K は本ノートの範囲では，$\mathbb{R}$ か $\mathbb{C}$ と考えてよい。）
$A[i,j]$	行列 A の i,j 成分
I	単位行列（すなわち $\mathrm{I}[i,j] = [i=j]$ である）
I_n	n 次単位行列（すなわち $\mathrm{I}_n = (\vec{e}_1, \ldots, \vec{e}_n)$ ）
O	零行列（すなわち $\mathrm{O}[i,j] = 0$ である）
$\mathrm{O}_{m\times n}$	m 行 n 列の零行列
$\mathrm{diag}(a_1, a_2, \ldots, a_n)$	対角成分が $a_1, a_2, \ldots, a_n$ である n 次対角行列。すなわち $\mathrm{diag}(a_1, a_2, \ldots, a_n)[i,j] = [i=j]a_i$
$A^\top$	行列 A の転置 (すなわち $A^\top[i,j] = A[j,i]$)
P_σ	置換行列（すなわち $\mathrm{P}_\sigma[i,j] = [i=\sigma(j)]$，$\mathrm{P}_\sigma\vec{e}_j = \vec{e}_{\sigma(j)}$）
$\langle \vec{x} \mid \vec{y} \rangle$	ベクトル $\vec{x}$ と $\vec{y}$ の内積 (混用の多い $(\vec{x},\vec{y})$ や $\langle \vec{x},\vec{y} \rangle$ などの記法は避ける)
$\mathrm{Span}(\vec{x}_1, \ldots, \vec{x}_r)$	有限個のベクトル $\vec{x}_1, \ldots, \vec{x}_r$ が生成する部分空間
$\mathrm{Span}(S)$	ベクトルの集合 S が生成する部分空間
$V_f(\lambda)$	固有値 λ に属する f の固有空間
$W_f(\lambda)$	固有値 λ に属する f の広義固有空間 (一般固有空間)
$\mathrm{Hom}(V,W)$	ベクトル空間 V からベクトル空間 W への線形写像の全体
$\mathrm{End}(V)$	ベクトル空間 V 上の線形写像の全体，すなわち $\mathrm{Hom}(V,V)$

ベクトルを表すのに $\mathbf{e}$ や $\mathbf{x}$ のような太字を用いている教科書も多いが，本稿では矢印つきの文字を用いる。

縦ベクトル（列ベクトル）は，紙面の節約のために，しばしば横ベクトル（行ベクトル）の転置として表記する。たとえば $\begin{pmatrix} a \\ b \end{pmatrix}$ は $(a,b)^\top$ と書くことが多い。

$\mathrm{Span}(\vec{x}_1, \ldots, \vec{x}_r)$ や $\mathrm{Span}(S)$ は，教科書のように $\langle \vec{x}_1, \ldots, \vec{x}_r \rangle$ や $\langle S \rangle$ と書く流儀もあるが，混乱しやすいので本稿では用いない。

アルゴリズム

$\alpha;\beta;\cdots$	処理 $\alpha,\beta,\ldots$ をこの順に順次行なうことを表す。
for α **do** β	α を満たす各要素について β を行なうことを表す。(下記参照)
while α **do** β	α が成り立っている間，β を繰り返し行なうことを表す。(下記参照)
repeat β **until** α	α が成り立つまで，β を繰り返し行なうことを表す。(下記参照)
if α **then** β	α が成り立っているときだけ，β を行なうことを表す。
if α **then** β **else** γ	α が成り立っているとき β を，そうでないとき γ を行なうことを表す。
return a	（関数定義内で有効）関数値を a として呼び出し元のプログラムに戻る。

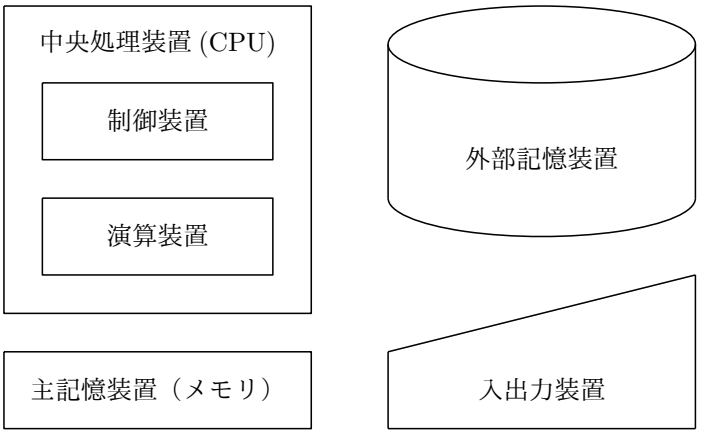
記法 **for** α **do** β において各要素の処理順は通常指定されないが，特に α が $i \in [n..m]$ の形で n，m が整数の場合は，$i = n$ の場合から始めて，$n \leqq m$ なら i を順次 1 づつ増やしながら，$n \geqq m$ なら 1 ずつ減らしながら，$i = m$ の場合まで，β を繰り返すことを表す。

while α **do** β では，初めから α が成り立っていなければ β が一度も実行されない。**repeat** β **until** α では，最初に β が実行されてから，α が判定されるので，少なくとも 1 回は β が実行される。

1 準備

1.1 コンピュータの基本構成

まず，基本知識として，コンピュータの構成と演算処理の流れについて概説する。標準的なコンピュータは，おおむね下図のような基本構成を持つ。



コンピュータでは，全ての情報を何らかの物理的な形態に表現せねばいかなる処理も不可能である。そのように表現された情報のかたまりをデータと呼ぶ。データの実際の表現形態は，磁気や電気の信号であったり，文字であったり，古くは紙テープや紙カード上の穴であったり，さまざまである。しかし，多くのデジタルコンピュータが共通に扱うデータの最小単位が，0 か 1 かのいわゆる 2 進数字であることは，既に長い伝統になっているし，今後もよほど特殊な事情が生じない限り，変わる事はないだろう。

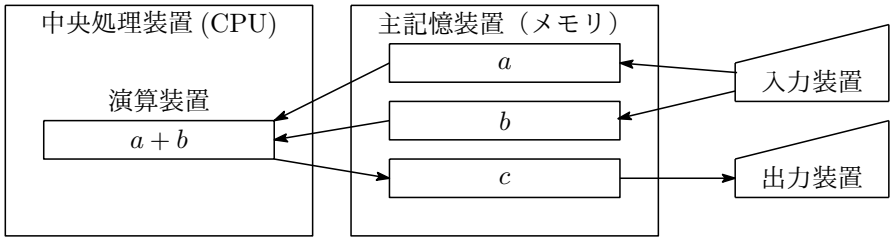
従って，コンピュータで何かを処理する場合，まず，情報を 2 進データに変換し，その 2 進データに様々な演算を行って得られた結果を，また人間にとって分かりやすい情報の形に変えなければならない。抽象的には，この外界における情報表現とコンピュータ内での 2 進表現との間の変換を行なう装置が，いわゆる**入出力装置**であると考えて，ほぼ間違いない。別の観点から言うと，入出力装置は，いわばコンピューターの目や手足に当たる装置であり，コンピュータが人間など外の世界との情報交換するときの主役である。データやプログラムは，既に外部記憶装置に納められいる場合を除けば，全て入力装置を経由してコンピュータに入って来る。また，計算結果なども，ディスプレイやプリンタなど，出力装置を通して人間に伝えることになる。

コンピュータの脳に当たるものが**中央処理装置 (CPU, Central Processing Unit)** である。CPU 自身が異なる機能を持ったいくつかの部分に分割されるが，その中で特に重要な役割を果たすのが制御装置と演算装置である。**制御装置**は，仕事を手順通り進めるために，各装置への指示や監督を行なう。**演算装置**は，加減乗除をはじめとして，データに対する実際の演算処理を行なう。

もちろん，脳以外の内臓に当たる装置も必要である。それにも様々なものがあるが，その代表として挙げられ，ほとんどどんなコンピュータにも備わっているのが**主記憶装置 (メモリ)** である。多くのコンピュータはそれとは別に外部記憶装置を伴うが，ここでは詳述しない。メモリは，文字通り，データを記憶・格納するための装置であり，制御装置の指示に従って，演算装置へデータを送り込んだり，逆に演算装置のデータを一時的に格納するという機能を持つ。また，処理を実際に監督するのは制御装置で

あるが，その処理手順を書いたデータは，制御装置が参照するためにメモリ上に格納される。この手順書をプログラムと呼ぶ。

さて，今 a，b という 2 つの数を足して，和 $a+b=c$ を計算するという簡単な仕事を考えよう。a と b という 2 つの数は，キーボードなどの入力装置から入力され，和 c は，例えばディスプレイから出力されるいうのは，上に述べた通りだが，これらのデータが直接に演算装置と入出力装置の間でやり取りされることは普通ない。通常は，下図のように入力された a や b は 2 進データとしていったんメモリに格納される。



メモリには，データを記憶する以外の機能はなく，普通，足し算を行なうことはできない。従って，$a+b$ を計算するには，まずデータ a が CPU 内の演算装置に送られる。続いてデータ b が送られ，演算装置上の a に足し込まれて $a+b$ が計算される。さらに，結果が c としてメモリに戻され，最終的に出力装置に表示または印刷されて人間に見える結果となる。その手順全体を監督しているのが CPU 内の制御装置である。また，この手順書自体も，プログラムとしてメモリ上の別の箇所に格納されている。

もっと複雑な計算も，基本的に，この入出力装置，メモリ，CPU という 3 者の役割分担によって遂行される。

1.2 漸近記法

本講義の主テーマは数学上の様々な計算を，実際に手やコンピュータープログラムによって，実行する際になるべく効率の良い手順 (アルゴリズム) を探し求め，どの程度の計算の手間でそのアルゴリズムが実行可能かを見積もることである。もちろん，扱うデータのサイズが大きくなれば，実行の手間 (実行に要する時間) も増えるのが当然なので，データサイズが大きくなったとき，実行時間がどのように増えていくかによって，通常はアルゴリズムの良し悪しを議論することになる。

このデータ量と計算時間の漸近的な関係を表現するために，ランダウが解析学のために導入した o 記法，O 記法，さらにはそれらを多少修正した θ 記法，Ω 記法，ω 記法などを用いる。

定義 1.1 (漸近記法)**.** $f(n)$ を $\mathbb{N}$ から $\mathbb{R}$ の関数とするとき，

$$\Theta(f(n)) \stackrel{\text{def}}{=} \{g(n) \mid \text{正の実数 } c_0, c_1 \text{ が存在して，十分大きな } n \in \mathbb{N} \text{ に対して } c_0 \leq \frac{g(n)}{f(n)} \leq c_1\}$$

$$O(f(n)) \stackrel{\text{def}}{=} \{g(n) \mid \text{正の実数 } c_1 \text{ が存在して，十分大きな } n \in \mathbb{N} \text{ に対して } 0 \leq \frac{g(n)}{f(n)} \leq c_1\}$$

$$o(f(n)) \stackrel{\text{def}}{=} \{g(n) \mid \lim_{n\to\infty} \frac{g(n)}{f(n)} = 0\}$$

$$\Omega(f(n)) \stackrel{\text{def}}{=} \{g(n) \mid \text{正の実数 } c_0 \text{ が存在して，十分大きな } n \in \mathbb{N} \text{ に対して } c_0 \leq \frac{g(n)}{f(n)}\}$$

$$\omega(f(n)) \stackrel{\text{def}}{=} \{g(n) \mid \lim_{n\to\infty} \frac{g(n)}{f(n)} = +\infty\}$$

たとえば，

$$10n^2 + 9n + 8 \in \Theta(n^2), \quad 1000n \in o(n^2), \quad 2 + (-1)^n \in \Theta(1),$$
$$1 + (-1)^n \notin \Theta(1), \quad 1 + (-1)^n \in O(1), \quad 1 + (-1)^n \notin o(1)$$

である。

問題 1.2. $\Theta(f(n)) \cup o(f(n)) \subset O(f(n))$ である。

問題 1.3. $\Theta(f(n)) = \Omega(f(n)) \cap O(f(n))$ である。

問題 1.4. $f(n), g(n), h(n)$ を $\mathbb{N}$ から $\mathbb{R}$ の関数とする。

(a) $f \in O(g)$ かつ $g \in O(h)$ ならば $f \in O(h)$ である。

(b) $f \in O(g)$ かつ $g \in o(h)$ ならば $f \in o(h)$ である。

(c) $f \in o(g)$ かつ $g \in O(h)$ ならば $f \in O(h)$ である。

$g(n) \in \Theta(f(n))$ や $g(n) \in O(f(n))$ などの代わりに $g(n) = \Theta(f(n))$，$g(n) = O(f(n))$ と書くことが多い。つまり，記法 $\Theta(f(n))$ や $O(f(n))$ は，集合というより，そこに属する関数の 1 つを代表するよ

うな意味合いでしばしば使う。次のような形でもよく使われる。

$$n^3 + 2n^2 + 3 = n^3 + o(n^3),$$
$$\Theta(f(n)) + O(f(n)) = \Theta(f(n)), \quad c\,\Theta(f(n)) = \Theta(f(n)),$$
$$O(f(n)) + O(f(n)) = O(f(n)), \quad c\,O(f(n)) = O(f(n)),$$
$$o(f(n)) + \Theta(f(n)) = \Theta(f(n)), \quad o(f(n))O(g(n)) = o(f(n)g(n))$$

上で c は正の定数とする。

問題 1.5. 上の 7 つの式の意味を考え，それらを証明せよ。

問題 1.6. $f(n), g(n)$ を $\mathbb{N}$ から $\mathbb{R}$ の関数とするとき，次を示せ。

(a) $f = O(g) \iff g = \Omega(f)$

(b) $f = \Omega(g) \iff g = \omega(f)$

(c) $f = \Theta(g) \iff g = \Theta(f)$

問題 1.7. $\max\{f(n), g(n)\} \in \Theta(f(n) + g(n))$ である。

問題 1.8. 次を正しいかどうかを判定せよ

$$2^{n+1} \in O(2^n), \qquad 2^{2n} \in O(2^n)$$

2 数値と数式の表現，基本演算アルゴリズム

2.1 数値の表現

コンピュータ内では，数値，文字を含めたあらゆるデータが，0 と 1 の列（2 進符号）の形に表現されていると考えてよい[1]。そこで，ここではまず，通常の数値を 2 進符号で表現する方法について説明する。

0 または 1 という値をとる最小のデータ単位を**ビット**（bit, binary digit）と呼ぶ。もちろん，通常の数値は 1 ビットでは表現不能であるから，1 つの数値を表すには何ビットかをまとめて利用することになる。データ処理の基本単位に多く使われるのが 8 ビットをまとめた**バイト**（byte）という単位であり，英語のアルファベットなど，通常の欧文文字は 1 バイト（たとえば ASCII コード）で表現される。これに対して，日本語は数千の漢字を使うので 2 バイト（16 ビット）のデータを 1 文字分として用いる。

1 バイトでは，$2^8 = 256$ 種類の情報しか表せないので，一般の数値を表すにもやや不足気味である。そこで，通常のコンピュータでは，1 つ整数を表すのに 32 ビットや 64 ビットのデータを用いる。これを**ワード**と呼ぶ。もちろん 10 進で 100 桁もあるような大きな数を表すには，これでも不足なので，このワードを単位として，それらを複数個組み合わせたデータを用いるが，ここで重要なのは，加算や乗算など 1 回の基本演算の単位となるのが，通常ワードであるということだ。ここでは 1 ワードのビット数を n として一般的に考える。先に述べたように n は 32 か 64 であることが多いが，コンピュータによって異なる。

計算が可能かどうかという観点だけから見れば，数値を表現するのにどのような符号を使うかは問題とならない。要求されることは，元の数値情報とデータ表現の間に確たる対応が取れていて，一方から他方が必ず復元できるということだけである。

しかし，計算の効率という観点から見ると，コンピュータ内部でのデータ表現をどうするかという問題は決してなおざりにできない。計算を速くするために，コンピュータ内部で情報をどのように保持するとよいかを考えることは，計算アルゴリズム設計の良し悪しに直結する重要な課題である。個々の問題毎に最適な表現があると考えられるが，ここでは一般に整数を表現するのに通常のコンピュータで取られている方法を概観しよう。

2.1.1 整数の表記（2 の補数）

周知のように 0 を含めた自然数を表現するには 2 進表記を用いる。では，負の数を含めた整数はどのように表記しているだろうか。素朴に考えると ± の符号部分と絶対値部分の 2 つのデータに分けて表現すると良さそうだが，四則演算の仕組みができるだけ簡単になるように，現代のコンピュータでは 2 の補数表記を用いるのが普通である。0 の周辺の整数の 2 の補数表記がどうなるかを示すと次ページの図 1 のようになる。

0 や 1 の左にある $\cdots$ はその数字が左に無限に続くことを表す。もちろん，コンピュータ内のデータの長さは有限だから，実際にはその列を適当なところで打ち切って格納することになる。1 つの数値を表すのに，1 ワード（n ビット）を用いるとすると，例えば $n = 8$ なら，-5 は 11111011，3 は 00000011

[1] この講義では，通常の文字を扱うようなアルゴリズムに触れることはほとんどないが，コンピュータが，普通にキーボードやディスプレイなどから読み書きする文字なども，ASCII コードや JIS コードといった 2 進符号，すなわち 0 と 1 の列で通常は表現している

2 の補数表記	数値
$\cdots 00101$	5
$\cdots 00100$	4
$\cdots 00011$	3
$\cdots 00010$	2
$\cdots 00001$	1
$\cdots 00000$	0
$\cdots 11111$	-1
$\cdots 11110$	-2
$\cdots 11101$	-3
$\cdots 11100$	-4
$\cdots 11011$	-5

図 1: 整数の 2 進表記（2 の補数による）

と表現する。符号と絶対値に分けて表現するより，2 の補数表記が優れている点は，0 の表記が $+0$ と -0 の 2 種類になったりしない点と，例えば加算のときに，符号によって操作を変えたりする必要がなく，そのまま単純に 2 進数の足し算を実行すればよい点などが挙げられる。乗算についても同様のことがいえる。

（右端を第 0 桁として左に数えたとき）通常の 2 進数の第 i 桁めの 1 は，2^i を表すと考えればいいが，n ビットで打ち切った 2 の補数表示の場合，左端の 1 だけは 2^{n-1} でなく -2^{n-1} を表すと考えるとよい。たとえば 11111011 は

$$-2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0 = -128 + 64 + 32 + 16 + 8 + 2 + 1 = -5$$

を表すというわけである。

2.1.2 実数の表記（浮動小数点数）

観測データなどをあつかう計算では，近似値でかまわないが，整数以外の数値を表現することが普通求められる。このためには，普通，一つの数値を符号，指数部，仮数部に分けて表現する。例えば $\frac{1}{10}$ であれば，

$$+\frac{1}{10} = +0.0001100110011001100\cdots = +2^{-4} \times 1.1001100110011001\cdots$$

つまり，通常の 2 進表記で表した上で，小数点位置を一番左の 1 のすぐ右に移動した表現を考える。そして，符号 $+$，指数部 -4，仮数部 $1001100110011001\cdots$ の 3 組で $\frac{1}{10}$ を表記するのである。もちろん，仮数部は一般には無限なので実際には適当なところで切り捨てる。この符号，指数部，仮数部をまとめ 1 ワードなり 2 ワードなりのデータとしたものを**浮動小数点数**（floating point number）と呼ぶ。

浮動小数点数の具体的な符号化の方法は，現実のコンピュータでもいくつかの方式が共存していて何が標準か定めにくく，本講義ノートでは当面必要がないので，ここには詳述しない。おさえておくべき重要な点は，仮数部ではほとんど必然的に下位ビットの切り捨て（または切り上げ）が起こっているので，ほぼ確実に誤差を含んでいるということだ。

グレイコード	数値
$\cdots 00000$	0
$\cdots 00001$	1
$\cdots 00011$	2
$\cdots 00010$	3
$\cdots 00110$	4
$\cdots 00111$	5
$\cdots 00101$	6
$\cdots 00100$	7
$\cdots 01100$	8
$\cdots 01101$	9
$\cdots 01111$	10

図 2: 整数の 2 進表記（グレイコード）

2.1.3 自然数の表記（グレイコード）★

整数のビット表現だけとっても，目的に合わせて様々な方式が考案されている。例えばグレイコードというものの最初の方の表は上の図 2 のようになる。

一般にグレイコード c が表す自然数を $\mathrm{gr}(c)$ と記すとしよう。このとき c を n 桁のグレイコードとすると，$n+1$ 桁のグレイコードが表す自然数は次のように帰納的に定義される。

$$\mathrm{gr}(0c) = \mathrm{gr}(c), \quad \mathrm{gr}(1c) = 2^{n+1} - 1 - \mathrm{gr}(c),$$

グレイコードは，加算をするには不向きな表現だが，隣り合った整数の表現が 1 ビットしか異ならないという特徴があり，アメリカでカウンタ設計のアイデアとして特許が取られたことで有名である。また，九連環（チャイニーズリング）と呼ばれる知恵の輪の解法との間の一見意外な関係が知られている。

問題 2.1. グレイコードと通常の 2 進表記との間の変換を行なう行なうアルゴリズムを与えよ。

問題 2.2. 九連環と呼ばれる知恵の輪について調べ，その解法とグレイコードとの関連について述べよ。

2.1.4 整数の表記（-2 進）★

負の整数を自然に表現できる記数法という意味では -2 進表記というものもある。0 の周辺の整数の -2 進表記は次ページの図 3 のようになる。

コードと数値の対応がどうなっているか一目では分からないだろうが，実は，第 i 桁目のビットが，2 進表記では 2^i を表すの対して，-2 進表記では $(-2)^i$ を表すというだけの違いなので，通常の 2 進表記とそれほど変わるものではない。例えば, -3 の -2 進表記が 1101 なのは，$-3 = -8+4+1 = (-2)^3+(-2)^2+(-2)^0$ であるという事実を反映している。-2 進表記は，2 の補数表示のように左に 1 が無限に続く表記などというものを考えなくとも，有限のビット列だけで正負の整数を全て一意に表現できるという特徴を持つ。-2 進表記された数同士の加算や乗算は，2 の補数表記の場合ほど分かりやすくないが，負の桁上がりというものを考えることで，容易に計算することが可能である。

−2 進表記	数値
101	5
100	4
111	3
110	2
1	1
0	0
11	−1
10	−2
1101	−3
1100	−4
1111	−5

図 3: 整数の 2 進表記（−2 進法）

問題 2.3. −2 進表記された 2 つの整数の間の加減乗除のアルゴリズムを考えよ。

2.1.5 多精度整数の表記と加算

科学上の実験から得られる数値などは，どうせ誤差を含んでいる。従って，大きな桁数を考えてもあまり意味がないので，その場合，2.1.2 節で述べた浮動小数点数を使えば，普通は十分である。しかし，例えば，公開鍵暗号に使われる整数など，応用によっては，きわめて大きな桁数の数を誤差なしで正確に扱わねばならないことがある。この場合には，1 ワードのデータでは情報を正確に表現しきれないので，複数ワードを用いて 1 つの数値を表現することになる。

まず非負の整数の場合だけを考えることにしよう。この場合，よく用いられる方法は，その数値が何ワード分のデータに格納されているかという長さを表すデータと数値自身を表現するデータ列の組を用いるというものだ。例えば

$$k, a_0, a_1, \ldots, a_{k-1}$$

というような合計 $k+1$ ワードからなるデータで 1 つの数値

$$\sum_{i=0}^{k-1} a_i \cdot 2^{ni}$$

を表す。n はワード長である。各 a_i は　0 以上 2^n 未満の整数を表すことができるので，このような $k+1$ ワードのデータで 0 以上 2^{nk} 未満の整数を正確に表すことが可能になる。このようなデータを**多精度整数**（multiprecision integer）と呼び，以下では $[a_0 \ \cdots \ a_{k-1}]$ と表記する。それが現している数値 $\sum_{i=0}^{k-1} a_i \cdot 2^{ni}$ と同一視することもあるので注意されたい。

一般に非負整数 $a \in \mathbb{N}$ を 2 進表記すると，その長さ（ビット長）は $\lceil \lg(a+1) \rceil$ となる。従って，1 ワードが n ビットのとき，a を正確に 2 進数として表現するには，少なくとも $\left\lceil \dfrac{\lg(a+1)}{n} \right\rceil$ 個のワードを必要とする。この数を $\mathrm{len}(a)$ と記し，a の**長さ**と呼ぶ。これに $\mathrm{len}(a)$ の値そのものを格納するワードを加えて，非負整数 a の表現には通常 $1+\mathrm{len}(a)$ 個のワードを用いる。

問題 2.4. 任意の正の整数 a，b について次を示せ。

$$\max\{\mathrm{len}(a), \mathrm{len}(b)\} \leqq \mathrm{len}(a+b) \leqq \max\{\mathrm{len}(a), \mathrm{len}(b)\} + 1$$
$$\mathrm{len}(ab) \leqq \mathrm{len}(a) + \mathrm{len}(b) \leqq \mathrm{len}(ab) + 1$$

問題 2.5. 一般に正の整数 n を r 進表記するとき，その桁数はいくつになるか？それを n，r，log などを用いて表現し，そのことを証明せよ。

次に多精度整数の加算について考える。どんなコンピュータ（の演算装置）も 1 ワードの 2 進数 2 つの加算を行なう機能を必ず備えているが，そのとき，足しあわされた結果が桁あふれを起こしたかどうかを，同時に検出することが普通は可能である。この機能の実現方法は，コンピュータの設計によるが，ここでは**補助演算装置**というものを用いることにする。加算の結果が桁あふれを起こしたときには，補助演算装置に 1 が格納され，演算装置には和の下位 n 桁だけが格納される。桁あふれを起こさなければ，補助演算装置には 0 が格納され，当然，演算装置には和の値が格納される。

すなわち，加えられる 2 つの数（1 ワード）を x，y とするとき，加算後の演算装置の値を z，補助演算装置の値を $\gamma \in \{0, 1\}$ とすると，

$$x + y = \gamma \cdot 2^n + z$$

が成り立つ。先の多精度整数の表記を流用してこの計算を

$$[z\ \gamma] \longleftarrow x + y$$

と表現しよう。

さて，2 つの多精度整数が $a = [a_0\ a_1\ \cdots\ a_{k-1}] = \sum_{i=0}^{k-1} a_i \cdot 2^{ni}$ と $b = [b_0\ b_1\ \ldots\ b_{k-1}] = \sum_{i=0}^{k-1} b_i \cdot 2^{ni}$ とが与えられているものとする。このとき，$a + b$ の計算を行ない，その結果 $c = [c_0\ c_1\ \cdots\ c_{k-1}] = \sum_{i=0}^{k-1} c_i \cdot 2^{ni}$ を得るアルゴリズムは次のようになる。

アルゴリズム（多精度整数の加算）

入力　多精度整数 $a = [a_0\ a_1\ \cdots\ a_{k-1}]$，$b = [b_0\ b_1\ \ldots\ b_{k-1}]$
出力　多精度整数 $c = [c_0\ c_1\ \cdots\ c_{k-1}]$（ただし $c = a + b$）

(1) $\gamma_0 \longleftarrow 0$
(2) **for** $i \in [0..k-1]$ **do** $[c_i\ \gamma_{i+1}] \longleftarrow a_i + b_i + \gamma_i$

上に述べた多精度整数とその加算アルゴリズムで重要なポイントは，通常，単精度整数の加算には単位時間がかかるので，多精度整数を加えるにはその長さ k に比例する時間，すなわち $\Theta(k)$ の時間を要するということだ。[2]

問題 2.6. 上では，a，b の表記の長さをどちらも k ワードとし，結果は最終的には桁あふれを起こさないものと仮定して，アルゴリズムを記述したが，a，b の表記の長さがを異なる場合や，最終結果が桁あふれを起こした場合に対しても動作するようにアルゴリズムを修正せよ。

絶対値の大きな負の数を正確に表す場合には，符号の情報をどこに持つかによって，いくつか流儀がありうるが，代表的なのは次の 2 通りだろう。1 つは長さの情報 k に符号情報を含める流儀である。も

[2] ということは，何進で表記しようと，その桁数に比例するということでもある。

う 1 つは 2 の補数を用いる流儀である。前者の方法は明らかだろうから，後者について簡単に述べると，その場合，最上位のワード a_{k-1} が符号の情報を含むことになる。表現する数値は非負整数だけの場合と同様で

$$\sum_{i=0}^{k-1} a_i \cdot 2^{ni}$$

であるが，a_{k-1} は 0 以上 2^n 未満の整数ではなく，-2^{n-1} 以上 2^{n-1} 未満の整数を表すことになり，結果として全体では -2^{nk-1} 以上 2^{nk-1} 未満の整数を表すことができる。

問題 2.7. コンピュータは，2 の補数を用いて多精度の負数を表すものとし，次のような単精度数の減算を行う機能を備えているとする。

$$[z\ \gamma] \longleftarrow x - y$$

ここで，$z \in [0..2^n-1]$，$\gamma \in \{-1, 0\}$ であり，$x-y = \gamma\cdot 2^n + z$ を満たす。この機能を用いて，(2 の補数で表示された) 2 つの多精度整数 $a = [a_0\ a_1\ \cdots\ a_{k-1}] = \sum_{i=0}^{k-1} a_i\cdot 2^{ni}$ と $b = [b_0\ b_1\ \ldots\ b_{k-1}] = \sum_{i=0}^{k-1} b_i\cdot 2^{ni}$ とが与えられたとき，$a-b$ を計算するアルゴリズムを与えよ。

2.2 多項式の表現

2.2.1 多項式の表現と加算

変数として $x, y, \ldots$，係数として可換環 R の要素を持つ多項式の全体を $R[x, y, \ldots]$ と記す。$R[x, y, \ldots]$ 上に自然に加減乗算が定義でき，$R[x, y, \ldots]$ 自身が可換環をなすことは，ほとんど自明であろう。

当面は，変数を x だけとし，係数を整数に限定した多項式，すなわち $\mathbb{Z}[x]$ の要素の表現とその上の演算について考える。コンピュータ上での整数の表現は既に論じたので，それは既知とする。

$\mathbb{Z}[x]$ の要素は，その係数を次数が低いほうから順に並べるだけで，表現できる。つまり，$(a_0, \ldots, a_n) \in \mathbb{Z}^n$ で多項式

$$a(x) = a_n x^n + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i$$

を表現する。つまり，次数 n の多項式 $a(x)$ は，コンピュータ上では $n+1$ 個の係数の組 $(a_0, \ldots, a_n)$ として表現されているものとする。

ここでは，表現を一意にするために $a_0 \neq 0$ と仮定し，$a_n x^n$ を多項式 $a(x)$ の**頭項** (leading term)，a_n を**頭係数** (leading coefficient) と呼び，それぞれ $\mathrm{lt}(a(x))$，$\mathrm{lc}(a(x))$ と記す。また，n を多項式 $a(x)$ の**次数** (degree) と呼び，$\deg(a)$ と記す。$\mathrm{lc}(a(x)) = 1$ である多項式は**モニック** (monic) であるという。

さて，2 つの多項式が $a(x)$ と $b(x)$ があり，それぞれ $(a_0, a_1, \ldots, a_k) \in \mathbb{Z}^k$ と $(b_0, b_1, \ldots, b_k) \in \mathbb{Z}^k$ の形に表現されているとする。すなわち，

$$a(x) = \sum_{i=0}^{k} a_i x^i, \qquad b(x) = \sum_{i=0}^{k} b_i x^i$$

である。このとき，$a(x) + b(x)$ の計算を行ない，その結果 $c(x) = \sum_{i=0}^{k} c_i x^i$ を得るアルゴリズムは次のようになる。

アルゴリズム（多項式の加算）

入力 多項式 $a(x)=\sum_{i=0}^{k} a_i x^i$，$b(x)=\sum_{i=0}^{k} b_i x^i$
出力 多項式 $c(x)=\sum_{i=0}^{k} c_i x^i$ $(c(x)=a(x)+b(x))$

(1) **for** $i \in [0..k]$ **do** $c_i \longleftarrow a_i + b_i$

上のアルゴリズムは，当然ながら多精度整数の加算のアルゴリズムに酷似しているが，繰り上がりの概念が不要なので，アルゴリズムとしてはさらに簡単なものになっている。しかし，大きな違いが表記 $c_i \longleftarrow a_i + b_i$ の中に隠れていることを指摘しておこう。これは，a_i と b_i の和を計算し c_i の値とすることを表しているが，a_i や b_i が多精度整数の場合は，当然，前節で述べた多精度整数の加算アルゴリズムを用いて計算することを意味する。このように，見た目の表現は，より簡単になっていても，実際には別のもっと複雑なアルゴリズムをサブルーチンとして利用することがあるので注意されたい。サブルーチンとは，別のアルゴリズム（プログラム）の下請けの形で利用されるアルゴリズム（プログラム）のことである。

上では，$a(x)$，$b(x)$ の次数をどちらも k として，アルゴリズムを記述したが，整数の場合の問題 2.6 と同様，$\deg(a) \neq \deg(b)$ の場合にも対応できるようにアルゴリズムを修正するのは容易だろう。

多精度整数の加算と同様，多公式の加算アルゴリズムで重要なポイントは，2 つの多項式の加算にかかる時間が，各係数の桁数の総和 $\sum_{i=0}^{k-1} \mathrm{len}(a_i)$ に比例するということだ。特に，係数がすべて単精度ならば，その時間は $\Theta(k)$ だし，係数がすべて m 桁の数ならば $\Theta(mk)$ である。

2.2.2 ホーナー法

多項式の乗算の話に進む前に，特定の値 $x = x_0$ に対して，多項式 $a(x)$ がとる値 $a(x_0)$ を計算するという問題に触れておく。例えば $a(x) = 5x^4 + 3x^3 - 2x^2 + 8x - 10$， $x = 10$ の場合，

$$a(10) = 5 \cdot 10^4 + 3 \cdot 10^3 - 2 \cdot 10^2 + 8 \cdot 10 - 10 = 52870$$

である。この値は，$x = 10$ だから，暗算でも計算できるが，x がもっと厄介な数のときには，素朴に計算を実行すると，$5x^4$ の計算に 4 回の乗算を要する。それ以外の項を計算するのにそれぞれ $3, 2, 1$ 回の乗算を要するので，乗算だけで計 10 回の計算を要する。一般に $\deg a = k$ であれば，素朴な方法では $k(k+1)/2$ 回の乗算と k 回の加（減）算を要する。

この計算は，多項式 $a(x)$ を

$$((\cdots(a_n x + a_{n-1})x + \ldots a_2)x + a_1)x + a_0$$

と考えることで，著しく簡単になる。例えば $a(x) = 5x^4 + 3x^3 - 2x^2 + 8x - 10$， $x = 10$ の場合，

$$a(10) = (((5 \cdot 10 + 3) \cdot 10 - 2) \cdot 10 + 8) \cdot 10 - 10$$

という式を括弧の指定通りに計算するのである。このとき，乗算の回数は 4 回である。一般に $\deg(a) = k$ であれば，上の素朴な方法では，乗算の回数は $\Theta(k^2)$ だが，この方法による乗算と加（減）算の回数はともに k で圧倒的に少ない。この方法はホーナー法と呼ばれ，x の特定の値に対して，多項式 $a(x)$ の値を計算する標準的な方法となっている。きちんとアルゴリズムの形に表現すると次のようになる。

アルゴリズム（ホーナー法による多項式の値の計算）

入力 多項式 $a(x) = \sum_{i=0}^{k} a_i x^i$，整数 x_0
出力 整数 $y = a(x_0)$

(1) $y \longleftarrow a_k$
(2) **for** $i \in [k-1..0]$ **do** $y \longleftarrow yx_0 + a_i$

$y \longleftarrow yx_0 + a_i$ の計算では，y，x_0，a_i が多精度整数であれば多精度整数の加減乗算を用いることは，言うまでもない。多精度整数の乗算の詳細については後述する。また，当面，整数以外には関心がないので，値 x_0 は整数と仮定したが， ホーナー法自体は x_0 が有理数や浮動小数点数の場合でも有効である。当然，その場合，$yx_0 + a_i$ の計算は，それぞれのデータ表現に合せて行なう必要がある。

問題 2.8. 係数 a_i と x をともに m 桁の整数とするとき，上のアルゴリズムの計算量を Θ 記法で表せ。

2.2.3 非負整数の 2 進 10 進変換

既知ではあろうが，非負整数の 2 進表記と 10 進表記の間の相互の変換について，ここでホーナー法との関連から少し触れておこう。

例えば 1011011 という 2 進表記を 10 進表記にすることを考える。右から i 桁目が 2^i を表していることから，この表記が表すのは，通常の 10 進表記では

$$2^6 + 2^4 + 2^3 + 2^1 + 2^0 = 64 + 16 + 8 + 2 + 1 = 91$$

であることが分かる。しかし桁数が大きくなると，この種の式を計算するのは容易ではなくなる。その場合にも，ホーナー法の考え方が有用である。すなわち，上の数は

$$(((((1 \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 1 = 91$$

としても計算できる。この方法で，一般に 2 進 k 桁の数を 10 進にするのに要する計算は，0 または 1 を足す操作を無視してもよいものと考えると，2 倍するという操作が $k-1$ 回だけである。

ホーナー法的考え方は，10 進表記を 2 進表記に変換するときには，さらに有用である。10 進表記 91 を逆に 2 進表記にする場合を例にすると，これは次のような計算によって通常行なう。

0	1	2	5	11	22	45	91
1	0	1	1	0	1	1	

この計算の上段右端が与えられた 10 進表記の数値である。上段の各数値の左の数値は，その数値を 2 で整除したときの商であり，その下段の数値がそのときの余りである。例えば，91 の左の 45 と 1 は，91 を 2 で割ると商は 45 で 1 余ることを表す。この計算を左端上段に 0 が現れるまで続けると，下段に求める 2 進表記がそのまま現れる。つまり，91 の 2 進表記は 1011011 だ。

上の計算がホーナー法による 2 進から 10 進の変換を完全に逆向きに辿っていることは，自明であろう。一般に 2 進表記が k 桁になる数値に対しては，2 で割って商と余りを取るという操作を k 回行なうことで 2 進表記への変換が完了する。

2.2.4 小数・分数の 2 進 10 進変換

これも既知ではあろうが，整数以外の数値についても，2 進表記と他の表記の間の相互変換法に触れておく。2.1.2 節で $\frac{1}{10}$ の 2 進表記が $0.0001100110011001100\cdots$ となることを用いているが，これを例に説明を試みよう。一般に有理数は，何進で表現しても，有限小数または循環小数になる。有限小数の場合は特に困難もないであろうから，2 進循環小数表現を分数表現に変換する方法について述べる。上の $0.0001100110011001100\cdots$ を循環小数としてコンパクトに表すと $0.0\dot{0}01\dot{1}$ である。数字の上の点が循環節の開始と終了を表す。さて，2 進表記において，小数点の右から第 n 桁目は 2^{-n} という数値を表すので，上の表記は

$$2^{-4}+2^{-5}+2^{-8}+2^{-9}+2^{-12}+2^{-13}+\cdots$$

という値を表す。$2^{-4}+2^{-5}$ でくくり，等比級数の和を計算すると，この値は

$$(2^{-4}+2^{-5})(1+2^{-4}+2^{-8}+\cdots)=\frac{3}{32}\cdot\frac{1}{1-2^{-4}}=\frac{3}{32}\cdot\frac{16}{15}=\frac{1}{10}$$

であることが確かめられる。しかし，等比級数を毎回計算するのは得策とはいえないので，通常は次のような方法をとる。まず循環節部分 $d_1d_2\ldots d_{k-1}d_k$ の通常の 2 進整数表記としての値を p とする。例えば，上の 0011 の場合，$p=3$ である。このとき，循環小数 $0.\dot{d_1}d_2\ldots d_{k-1}\dot{d_k}$ が $\frac{p}{2^k-1}$ を表すことは容易に示される。例えば，$0.\dot{0}01\dot{1}$ は $\frac{3}{2^4-1}=\frac{1}{5}$ を表す。従って，小数点を循環節のすぐ左にずらすことで，比較的容易にそれが表す有理数を計算することができる。例えば，

$$0.0\dot{0}01\dot{1}=2^{-1}\times 0.\dot{0}01\dot{1}=\frac{1}{2}\times\frac{1}{5}=\frac{1}{10}$$

である。別の例を挙げると $0.1010110110110\cdots$ ならば

$$0.1010110110110\cdots=2^{-2}\times 10.\dot{1}0\dot{1}=\frac{1}{4}\times\left(2+\frac{5}{2^3-1}\right)=\frac{1}{4}\times\frac{19}{7}=\frac{19}{28}$$

である。

問題 2.9. 2 進循環少数 $10.1\dot{0}11\dot{0}$ を分数として表せ。分子と分母は通常の 10 進整数として表記すること。

次に，逆に整数以外の数値を 2 進表記に変換する方法を与えよう。一般に数値 x は整数部分 $\lfloor x\rfloor$ と小数部分 $\{x\}=x-\lfloor x\rfloor\in[0..1)$ に分けられる。x の 2 進表記は，$\lfloor x\rfloor$ の 2 進表記 z と $\{x\}$ の 2 進表記 p とを合せて $z.p$ とすれば得られるから，$y\in[0..1)$ なる数値 y を 2 進表記に変換できればよい。この変換は，やはりホーナー法的考え方でやるのがいい。$\frac{1}{10}$ を例に説明しよう。

1/10	1/5	2/5	4/5	3/5	1/5	2/5	4/5	3/5	$\cdots$
	0	0	0	1	1	0	0	1	$\cdots$

上の計算で左端の数値が変換したい数 y である。上の段の各数値の右にはそれを 2 倍した数値が書かれている。ただし，その結果が 1 以上なら，それから 1 を引いた数を書き，その下に 1 を書く。1 未満ならそのままで，下には 0 を書く。この操作を繰り返すと，下の段に y の 2 進表記の小数点より右側

が現れる。上の例だと $0.00011001\cdots$ である。結果が循環小数になる場合，上の段に同じ数値が 2 度現れる。上の例では，1/5 が 2 度出現しているが，2 度目の出現以降は同じ計算が繰り返されるので，2 進表記も循環し $0.0\dot{0}01\dot{1}$ になることが分かる。$\dfrac{19}{28}$ の例を計算すると

$$\begin{array}{cccccccc} 19/28 & 5/14 & 5/7 & 3/7 & 6/7 & 5/7 & \cdots \\ & 1 & 0 & 1 & 0 & 1 & \cdots \end{array}$$

である。右端上段 5/7 の出現は，2 度目だからこの後の計算は繰り返しとなる。よって 19/28 の 2 進表記は $0.10\dot{1}0\dot{1}$ である。この計算法は，有理数の場合に限らず有効である。例えば円周率 $3.141592\cdots$ の場合，近似値 3.1416 を用いると

$$\begin{array}{cccccccc} .1416 & .2832 & .5664 & .1328 & .2656 & .5312 & .0624 & \cdots \\ & 0 & 0 & 1 & 0 & 0 & 1 & \cdots \end{array}$$

より，小数部分 .1416 の 2 進表記は $.001001\cdots$ となる。よって，3 の 2 進表記 11 と併せて，円周率の 2 進表記は $11.001001\cdots$ となることが分かる。

問題 2.10. 次の数を 2 進表記せよ。負の数は整数部 8 桁の 2 の補数で表記せよ。有理数の場合，有限小数または循環小数として，また無理数は (9 桁目を 0 捨 1 入して)，小数点以下 8 桁の数として表せ。なお，問題中の数値はすべて 10 進表記である。

(a) -90.5
(b) $-1/5$
(c) 2 の平方根 $\sqrt{2}$ （約 1.41421356）

問題 2.11. 2 進表記された次の数を 10 進表記に戻せ。循環少数は分数に戻すこと。

(a) 10101010101
(b) $1010.1\dot{0}11\dot{0}$

問題 2.12.
自然対数の底 $e = 2.718281828\cdots$ を 2 進表記で（11 桁目を 0 捨 1 入して）小数点以下 10 桁まで求めよ。

2.3 乗算

次に多精度整数や多項式の乗算について考える。

2.3.1 多項式の乗算

加算もそうだったが，多項式には繰り上がりの問題がなく，技術的には多精度整数より簡単なので，そちらを先に考える。多項式 $a(x)$ と $b(x)$ の乗算を行なう素朴なアルゴリズムは次のようなものである。

アルゴリズム（多項式の乗算）

入力　多項式 $a(x)=\sum_{i=0}^{k} a_i x^i$，$b(x)=\sum_{i=0}^{m} b_i x^i$
出力　多項式 $c(x)=\sum_{i=0}^{k+m} c_i x^i$ $(c(x)=a(x)b(x))$

(1) **for** $i \in [0..k+m]$ **do** $c_i \longleftarrow 0$
(2) **for** $i \in [0..k]$ **do for** $j \in [0..m]$ **do** $c_{i+j} \longleftarrow c_{i+j} + a_i b_j$

このアルゴリズムは c の係数 c_l が畳み込み $\sum_{l=i+j} a_i b_j$ で与えられることを直接表現したものである。当然だが，$c_{i+j} \longleftarrow c_{i+j} + a_i b_j$ の計算には，a_i と b_j が多精度整数であると，後述する多精度整数の乗算が必要になる。

2.3.2　多精度整数と単精度整数の乗算

さて多精度整数の乗算について検討しよう。一般に n 桁の数を 2 つ掛け合わせた結果は $2n$ 桁になる。従って，1 ワードの 2 進数 2 つの乗算結果を正確に格納するには 2 ワード分の記憶領域が必要である。ここでは補助演算装置にその機能を持たせることにする。つまり，一般に計算結果は，下位 n 桁だけが演算装置に格納され，上位桁は補助演算装置に格納されるものとする。

より精密に言うと，3 つの単精度数（1 ワード）を x，y，z とするとき，

$$xy + z = \gamma \cdot 2^n + w$$

なる w と γ の値をそれぞれ演算装置と補助演算装置に格納する仕組みがコンピューターには組み込んであるものとする。多精度整数の表記を流用して，この計算を

$$[w\ \gamma] \longleftarrow xy + z$$

と表現しよう。

さて，上の機能を用いて，多精度の整数と単精度の整数の乗算アルゴリズムを書くと次のようになる。

アルゴリズム（多精度整数と単精度整数の乗算）

入力　多精度整数 $a=[a_0\ a_1\ \cdots\ a_{k-1}]$，単精度整数 $b \in [0..2^n-1]$
出力　多精度整数 $c=[c_0\ c_1\ \cdots\ c_k]$（ただし $c=ab$）

(1) $\gamma_0 \longleftarrow 0$
(2) **for** $i \in [0..k-1]$ **do** $[c_i\ \gamma_{i+1}] \longleftarrow a_i b + \gamma_i$
(3) $c_k \longleftarrow \gamma_k$

このアルゴリズムは，通常の筆算による k 桁掛ける 1 桁の乗算を模しているだけだから，正しいことは明らかであろう。厳密には，m に関する帰納法により，

$$\gamma_m 2^{nm} + \sum_{i=0}^{m-1} c_i 2^{ni} = \left(\sum_{i=0}^{m-1} a_i 2^{ni} \right) b$$

証明すればよい。$m=k$ のとき，$\gamma_k = c_k$ だから $c=ab$ である。

問題 2.13. 上の式 $\gamma_m 2^{nm} + \sum_{i=0}^{m-1} c_i 2^{ni} = \left(\sum_{i=0}^{m-1} a_i 2^{ni}\right) b$ を証明せよ。

解答例 m に関する帰納法による。$m = 0$ のときは，右辺 $= 0 =$ 左辺 より明らかである。$m+1$ のときは，m の場合を用いて，

$$\begin{aligned}
\text{右辺} &= \left(\sum_{i=0}^{m} a_i 2^{ni}\right) b = a_m 2^{nm} b + \left(\sum_{i=0}^{m-1} a_i 2^{ni}\right) b = a_m 2^{nm} b + \gamma_m 2^{nm} + \sum_{i=0}^{m-1} c_i 2^{ni} \\
&= (a_m b + \gamma_m) 2^{nm} + \sum_{i=0}^{m-1} c_i 2^{ni} = (\gamma_{m+1} 2^n + c_m) 2^{nm} + \sum_{i=0}^{m-1} c_i 2^{ni} \\
&= \gamma_{m+1} 2^{n(m+1)} + \sum_{i=0}^{m} c_i 2^{ni} = \text{左辺}
\end{aligned}$$

である。 □

2.3.3 多精度整数と多精度整数の乗算

上の多精度整数と単精度整数の乗算をサブルーチンに用いれば，多精度整数同士の乗算も次のように容易に書ける。

アルゴリズム（多精度整数と多精度整数の乗算）

入力 多精度整数 $a = [a_0\ a_1\ \cdots\ a_{k-1}]$，$b \in [b_0\ b_1\ \cdots\ b_{m-1}]$
出力 多精度整数 $c = [c_0\ c_1\ \cdots\ c_{k+m-1}]$ （ただし $c = ab$）

(1) $[c_0\ \cdots\ c_k] \longleftarrow ab_0$
(2) **for** $i \in [1..m-1]$ **do** $[c_i\ \cdots\ c_{k+i}] \longleftarrow [c_i\ \cdots\ c_{k+i-1}] + ab_i$

このアルゴリズムも，通常の筆算による k 桁掛ける m 桁の乗算を模しているだけだから，正しいことは明らかであろう。当然であるが，ab_0 や ab_i の計算には多精度整数と単精度整数の乗算アルゴリズムが用いられ，$[c_i\ \cdots\ c_{k+i-1}] + ab_i$ の計算には多精度整数と多精度整数の加算アルゴリズムが用いられる。

2.4 剰余つき除算

コンピュータで行なう代数計算で，加減乗算の次にもっとも基本的なものは，剰余つきの除算である。すなわち与えれた整数 a と正の整数 b に対して

$$a = qb + r, \qquad 0 \leqq r < b$$

を満たす整数 q と r を求める計算である。a を被除数，b を除数，q を商，r を剰余と呼ぶ。除数 b が負の数の場合に商や剰余をどう定義するかはいくつか流儀があるから，ここでは触れないで，$b > 0$ を仮定する。普通の記法に従って，商 q を $a \div b$，剰余 r を $a \bmod b$ と記す。通常，q と r の計算は同時に行なうことが可能だから，そのような計算を

$$(q, r) \longleftarrow a \div b$$

と記す。

2.4.1 多項式の剰余つき除算

2 つの多項式 a と b についても同様の除算が定義できる。すなわち，

$$a(x) = q(x)b(x) + r(x), \qquad \deg r(x) < \deg b(x)$$

を満たす $q(x)$ と $r(x)$ を求める計算である。ただし，当面，多項式の係数の範囲を整数にしておきたいので，$b(x)$ はモニック，すなわち $\mathrm{lc}(b(x)) = 1$ と仮定する。記法

$$a(x) \div b(x), \quad a(x) \bmod b(x), \quad (q(x), r(x)) \longleftarrow a(x) \div b(x)$$

も同様に用いる。

問題 2.14. 上の定義による剰余つき除算の商と剰余は，整数と多項式のいずれの場合についても，被除数と除数が与えられれば一意に定まることを示せ。

与えられた多項式 $a(x) = \sum_{i=0}^{k} a_i x^i$ を被除数，与えられたモニックな多項式 $b(x) = \sum_{i=0}^{m} b_i x^i$ $(b_m = 1)$ を除数として，剰余つき除算を行なうアルゴリズムは次のようになる。

アルゴリズム（多項式の剰余つき除算）

入力 多項式 $a(x) = \sum_{i=0}^{k} a_i x^i$，$b(x) = \sum_{i=0}^{m} b_i x^i$ $(b_m = 1)$
出力 多項式 $q(x) = \sum_{i=0}^{k-m} q_i x^i$，$r(x) = \sum_{i=0}^{m-1} r_i x^i$ （ただし $a(x) = q(x)b(x) + r(x)$）

(1) **for** $i \in [k..0]$ **do** $r_i \longleftarrow a_i$ （i.e. $r(x) \longleftarrow a(x)$）
(2) **for** $i \in [k-m..0]$ **do**
　　$q_i \longleftarrow r_{i+m}$;
　　for $j \in [m..0]$ **do** $r_{i+j} \longleftarrow r_{i+j} - q_i b_j$;

上のアルゴリズムも通常の筆算による計算を模したものに過ぎないので，正しいことは明らかであろう。

問題 2.15 (1 次式による除算)**.** 上のアルゴリズムは， $b(x)$ が 1 次式 $x - c$ の場合，単に

$$q_k \longleftarrow a_k; \quad \textbf{for } i \in [k-1..0] \textbf{ do } q_i \longleftarrow a_i - q_{i+1}c;$$

と簡単化でき，商 $q(x)$ は $\sum_{i=0}^{k-1} q_{i+1}x^i$，剰余 r は q_0 として得られることを示せ。このとき，$r = q_0$ は $a(c)$ に等しいので，このようにして $a(x)$ を $x - c$ で割ったときの商 $q(x)$ と剰余 $r = a(c)$ を同時に求める方法を**組み立て除法**という。高校数学で習った人もいるだろう。ホーナー法で $a(c)$ を単独に求める場合と計算量を比較せよ

問題 2.16 (体上の多項式の除算)**.** 上では $b(x)$ がモニックであること仮定したが，$a(x)$，$b(x)$ が一般の体 K 上の多項式，すなわち $a(x), b(x) \in K[x]$ であるとき，$b(x) \neq 0$ なら同様の剰余つき除算がいつでも定義できることを示せ。また，任意の $a, b \in K$ について，加減乗除 $a+b$，$a-b$，ab，a/b $(b \neq 0)$ の計算は自由にできると仮定して，$K[x]$ 上の剰余つき除算を行なうアルゴリズムを与えよ。

問題 2.17 (一般の環上の多項式の除算)**.** $a(x)$，$b(x)$ が一般の可換環 R 上の多項式のとき，同様の剰余つき除算が定義できるための係数の条件を考えよ。また，その条件が満たされた場合に，$R[x]$ 上の剰余つき除算を行なうアルゴリズムを与えるには，R 上の演算としてはどういうものが計算できる必要があるか。

2.4.2 多精度整数の剰余つき除算

$0 \leqq a_1 < b$ のとき，倍精度整数（長さ 2 の多精度整数）$[a_0\ a_1]$ を単精度整数 b で割った商 q と剰余 r は，どちらも単精度で表現できる。コンピュータは，通常基本的な演算機能として，このような計算

$$(q, r) \longleftarrow [a_0\ a_1] \div b$$

を行い 2 つの単精度整数 q と r を返す機能を備えている。

問題 2.18. 上の機能を用いて，多精度整数を単精度整数で割る場合の剰余つき除算のアルゴリズムを与えよ。つまり，入力は被除数としての多精度整数と除数としての単精度整数であり，それに対して商として多精度整数，剰余として単精度整数を出力するアルゴリズムである。

多項式の場合と同様に，一般の多精度整数同士の剰余つき除算を行なうことも可能である。加算や乗算と同様，筆算をなぞったようなアルゴリズム化が可能であるが，多精度整数の除算の場合，繰り上がりの問題があるので，商の決定に試し割り操作が加わり，多項式同士の除算よりは複雑になる。そのため，ここには詳述しない。

3 整数の基本的性質 ☆

本章では整除性を中心に整数の基本的性質について述べる。また正の整数 n を法とした合同関係について論じ，剰余類の概念を導入する。中国剰余定理，オイラーの ϕ 関数，フェルマの小定理，平方剰余，メビウスの反転公式などについて学ぶ。

3.1 整除性 ☆

人間が生まれて最初に学ぶ数の概念は自然数（非負の整数）であろうし，それに負の数を加えた整数は，小中学校以来，もっとも馴染み深い数学概念であろうから，加減乗除などの一部の演算とそれに関わる多くの性質は既知のものとする。例えば，$(a+b)c = ac+bc$，$ab = ba$ や $ab = 0 \iff a = 0 \vee b = 0$ などの性質がある。特に最後のものは，$\mathbb{Z}$ が整域であること述べており，それから相殺法則 $ac = bc \Longrightarrow a = c$ などが出てくる重要なものであるが，小中学校以来，整数というものを学んで来た人には，当たり前としか思えないであろうから，以下では証明なしに用いる。

とは言っても，一部の記法には本講義ノート独特のものがあるし，一見当たり前に見えることでも，復習を兼ねて整理しておくべきことがいくつかあるので，それから始める。

3.1.1 整除性と素数 ☆

定義 3.1. 整数 $a \in \mathbb{Z}$ が別の整数の $b \in \mathbb{Z}$ の**約数**（divisor）であるとは，整数 $z \in \mathbb{Z}$ が存在して，$az = b$ であることをいう。逆に b を a の**倍数**（multiple）と呼ぶ。特に $a > 0$ のとき， $a \setminus b$ という記法を用い，a が b を**割り切る**という。$a \leqq 0$ のときは，通常，この記法は用いないが，逆に a が正の数ならば整数でなくとも，b/a が整数になることをいうために $a \setminus b$ と記すことがある。例えば $\pi \setminus -2\pi$ である。

定理 3.2. 任意の $a, b, c \in \mathbb{Z}$ について，次が成り立つ。

(a) $a \setminus a$，$1 \setminus a$，$a \setminus 0$;
(b) $a \setminus b \iff a \setminus -b$;
(c) $a \setminus b \wedge a \setminus c \Longrightarrow a \setminus (b \pm c)$;
(d) $a \setminus b \wedge b \setminus c \Longrightarrow a \setminus c$;
(e) $a \setminus b \wedge b \setminus a \iff a = b$ （特に $a \setminus \pm 1 \iff a = 1$）;

証明: どれも定義からの簡単な帰結である。 □

問題 3.3. 上の定理を証明せよ。

定義 3.4. n を 2 以上の整数とする。n は，1 と n 以外に正の約数を持たなければ**素数**（prime number）と呼び，そうでないとき**合成数**（composite number）と呼ぶ。

素因数分解の存在と一意性，すなわち次の内容を**算術の基本定理**という。

定理 3.5 (算術の基本定理)**.** 0 でないすべての整数 n は

$$n = \mathrm{sgn}(n) p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$$

と書ける。ここで，$\mathrm{sgn}(n)$ は n の符号，すなわち n の正負に応じて ± 1 であり，$p_1, \ldots, p_r$ は互いに異なる素数，$e_1, \ldots, e_r$ は正の整数である。さらに，この表記は素数の並べ替えを除けばただ 1 通りである。

証明: 素因数分解の存在，すなわち定理の前半部は明らかであろう。定理の後半部，すなわち素因数分解の一意性は，整数では剰余つき除算が可能なことからの帰結となる。以下でそれを示す。 □

問題 3.6. 素因数分解の存在，すなわち上の定理の前半部を証明せよ。

定理 3.7. 素数は無限個存在する。

証明: 背理法による。素数が有限個だとし，その最大のものを p とする。$p! + 1$ は 2 以上 p 以下のどんな正の整数でも割り切れない。したがって $p! + 1$ を割り切る素数は p より大きく，p を最大の素数としたことに矛盾する。 □

定理 3.8 (剰余つき除算)**.** $a, b \in \mathbb{Z}$ で $b > 0$ とする。このとき，次を満たす $q, r \in \mathbb{Z}$ がただ 1 組存在する。

$$a = qb + r, \quad 0 \leqq r < b$$

証明: 集合 $S = \{a - bq \in \mathbb{N} \mid q \in \mathbb{Z}\}$ を考える。$b > 0$ であるから，S は空でなく，$\mathbb{N}$ の性質により，最小元 r を持つ。定義により a は $bq + r$ の形に書け，もし $r \geqq b$ ならば $r' = r - b = a - (q + 1) \in S$ かつ $r' < r$ だから，r の最小性に反する。よって，$0 \leqq r < b$ である。

他に $a = q'b + r', 0 \leqq r' < b$ なる q', r' があるとすると，$(q'b + r') - (qb + r) = (q' - q)b + (r' - r) = 0$ なので，$r' - r$ は b で割り切れなくてはならないが，$|r' - r| < b$ だから $r' = r$ である。さらに $b > 0$ だから $(q - q')b = r' - r = 0$ より，$q = q'$ も導かれる。 □

定義 3.9. 上記定理における q は切り下げ関数を利用して $\lfloor a/b \rfloor$ と書けるが，これを q を**整商**（quotient）と呼ぶ。また，r は $a - \lfloor a/b \rfloor b$ と書けるが，$a \bmod b$ と表記して，b を**法とする**（modulo） a の**剰余**（residue, remainder）と呼ぶ。

これをさらに一般化して，a, b が必ずしも整数でなく，$a, b \in \mathbb{R}$ のときにも $a - \lfloor a/b \rfloor b$ を $a \bmod b$ と記す。

問題 3.10. 剰余の値は好きなだけずらすことができる。すなわち $a, b, x \in \mathbb{R}$ で $b > 0$ とするとき，次のような $q \in \mathbb{Z}$，$r \in \mathbb{R}$ がただ 1 組存在する。

$$a = qb + r, \quad x \leqq r < x + b$$

問題 3.11. $a, b, d \in \mathbb{Z}$ とする。このとき，$a \setminus b \iff da \setminus db$ である。

問題 3.12. n を合成数とする。$\sqrt{n}$ 以下の素数 p が存在して $p \setminus n$ である。

問題 3.13. m を正の整数とする。x を正の実数とするとき，区間 $[1..x]$ の中には m の倍数はいくつあるか？

問題 3.14. 任意の $x \in \mathbb{R}$ について $2\lfloor x \rfloor \leqq \lfloor 2x \rfloor \leqq 2\lfloor x \rfloor + 1$ である。

問題 3.15. 任意の $x \in \mathbb{R}$ と任意の正の整数 a, b, c に対して $\lfloor \lfloor x \rfloor / a \rfloor = \lfloor x/a \rfloor$，$\lfloor \lfloor x/a \rfloor / b \rfloor = \lfloor x/ab \rfloor$ である。

3.1.2 最大公約数 ☆

定義 3.16. $a, b \in \mathbb{Z}$ とする。d が a と b の共通の約数であれば，a と b の**公約数**（common divisor）という。さらに d が正で，他のすべての公約数の倍数なら**最大公約数**（GCD, greatest common divisor）という。a と b の最大公約数は，ただ 1 つ存在する。それを $\gcd(a, b)$ と記す。$\gcd(a, b) = 1$ のとき，a と b は**互いに素** (mutually prime, relatively prime) といい，$a \perp b$ と記す。

定理 3.17. $a, b \in \mathbb{Z}$ とする。このとき $s, t \in \mathbb{Z}$ が存在して $sa + tb = \gcd(a, b)$ と書ける。特に $a \perp b$ ならば $s, t \in \mathbb{Z}$ が存在して $as + bt = 1$ となる。

定理 3.17 は，以後の理論の根幹をなす重要な定理だが，その証明は，このような s と t の計算方法とともに，拡張ユークリッド互除法として 4.3 節で与える。

系 3.18. $a, b, r \in \mathbb{Z}$ とする。このとき $\gcd(a, b) \setminus r$ であるとき，かつそのときに限り，$s, t \in \mathbb{Z}$ が存在して $sa + tb = r$ と書ける。

証明: $sa + tb = r$ と書けるなら，$\gcd(a, b)$ は a と b の両方を割り切るので，$sa + tb = r$ を割り切る。

逆に $q \gcd(a, b) = r$ とする。前定理より $s'a + t'b = \gcd(a, b)$ となる s' と t' があるので，$s = qs'$, $s = qt'$ とおけば $r = sa + tb$ である。 □

補題 3.19. 任意の $a, b, c \in \mathbb{N}$ について，次が成り立つ

(a) $\gcd(a, b) = a \iff a \setminus b$
(b) $\gcd(a, 0) = \gcd(a, a) = \gcd(a, -a) = |a|$ かつ $\gcd(a, 1) = \gcd(a, -1) = 1$
(c) $\gcd(a, b) = \gcd(b, a)$ （可換律）
(d) $\gcd(a, \gcd(b, c)) = \gcd(\gcd(a, b), c)$（結合律）
(e) $\gcd(ca, cb) = |c| \gcd(a, b)$（分配律）
(f) $|a| = |b| \Longrightarrow \gcd(a, c) = \gcd(b, c)$

問題 3.20. 上の補題を証明せよ。

定理 3.21. $a, b, c \in \mathbb{Z}$ とする。このとき $c \setminus ab$ かつ $a \perp c$ ならば $c \setminus b$ である。

証明: $a \perp c$ だから $sa + tc = 1$ となる s, t が存在する。b 倍すると $sab + tcb = b$ だが，$c \setminus ab$ だから左辺は c で割り切れる。ゆえに $c \setminus b$ である。 □

定理 3.22. p を素数とし，$a, b \in Z$ とする。このとき，$p \setminus ab$ ならば $p \setminus a$ または $p \setminus b$ である。さらに一般に $a_1, \ldots, a_k \in Z$ に対して，$p \setminus a_1 \cdots a_k$ ならば $p \setminus a_1, \cdots, p \setminus a_k$ のいずれかが成り立つ。

証明: $p \setminus ab$ とする。$p \setminus a$ でないとすると，p は素数だから $\gcd(p, a) = 1$ であり，定理 3.21 より $p \setminus b$ である。一般の場合への拡張は容易であろう。 □

3.1.3 素因数分解の一意性 ☆

以上の準備の下で算術の基本定理（定理 3.5）の後半は次のように証明される。

証明: ある $n \in \mathbb{Z}$ が $\mathrm{sgn}(n) p_1^{e_1} p_2^{e_2} \ldots p_r^{e_r}$ と $\mathrm{sgn}(n) q_1^{f_1} q_2^{f_2} \ldots q_s^{f_s}$ とに 2 通りに素因数分解されたとする。素数は並べ替えてもいいので，$p_1 < \cdots < p_r$，$q_1 < \cdots < q_s$ と仮定する。$\mathrm{sgn}(n)$ はもちろん一致

しているので，他の項も一致していることを $r+s$ に関する帰納法で示す。$r=s=0$ ならば明らかである。r と s の一方は 0 でないとする。このとき明らかに他方も 0 でない。$p_1=q_1$，$e_1=f_1$ ならば，先頭の項を相殺して，$p_2^{e_2}\dots p_r^{e_r}=q_2^{f_2}\dots q_s^{f_s}$ に帰着されるが，帰納法の過程より，この両辺は一致する。$p_1=q_1$，$e_1<f_1$ なら，同様に相殺して，$p_2^{e_2}\dots p_r^{e_r}=q_1^{f_1-e_1}q_2^{f_2}\dots q_s^{f_s}$ に帰着される。$p_1=q_1$，$e_1>f_1$ の場合も，同様である。$p_1<q_1$ ならば，定理 3.22 より $p_1\setminus q_1,\dots,p_1\setminus q_s$ のどれかが成り立たねばならないが，$p_1<q_1<\cdots<q_s$ であり，p_1 も q_j も素数だからこれは不可能である。$p_1>q_1$ の場合も同様だ。 □

問題 3.23. 2 つの整数が互いに素であるための必要十分条件は，両方を割り切る素数が存在しないことである。

問題 3.24. p を素数とし，$k\in\mathbb{Z}$ を $0<k<p$ とする。このとき 2 項係数

$$\binom{p}{k}=\frac{p!}{k!(p-k)!}$$

は p の倍数になる。

問題 3.25.
整数は 1 以外のどんな平方数の倍数でもないとき，**無平方** (square free) と呼ぶ。次を証明せよ。

(a) 整数 n が無平方であるための必要十分条件は，n が異なる素数 p_i の積 $\mathrm{sgn}(n)p_1p_2\cdots p_r$ の形に素因数分解されることである。
(b) 任意の正の整数 n は $n=ab^2$ の形に因数分解し，a を無平方であるようにできる。

解答例
(a) n の素因分解 $\mathrm{sgn}(n)p_1p_2\cdots p_r$ の中に同じ素数 $p_i=p_j=p$ $(i\neq j)$ が現れるなら，n は明らかに平方数 p^2 $(\neq 1)$ の倍数だから無平方ではない。逆に n が無平方で無いとして，$b^2c=n$ とする。b の素因子を一つを p とすると，p^2 は n を割り切るので，素因数分解の一意性より，p は n の素因数分解 $\mathrm{sgn}(n)p_1p_2\cdots p_r$ の中に 2 回以上現れる。
(b) n の素因分解を $\mathrm{sgn}(n)p_1^{e_1}p_2^{e_2}\cdots p_r^{e^r}$ とする。ただし，各 p_i は異なる素数である。$f_i=e_i \bmod 2$，$g_i=(e_i-f_i)/2$ とし，$a=\mathrm{sgn}(a)p_1^{f_1}p_2^{f_2}\cdots p_r^{f^r}$，$b=p_1^{g_1}p_2^{g_2}\cdots p_r^{g^r}$ とすれば，明らかに a は無平方であり，$n=ab^2$ である。 □

問題 3.26. a,b を正の整数とする。写像 $\tau:[0..a-1]\times[0..b-1]\to[0..ab-1]$ を

$$\tau(s,t)\stackrel{\mathrm{def}}{=}(as+bt) \bmod ab$$

で定義するとき，τ が全単射であるための必要十分条件は $a\perp b$ である。

問題 3.27. a,b,c を正の整数とし，$a\perp b$ かつ $c\geqq(a-1)(b-1)$ とする。このとき非負の整数 s,t が存在して $c=as+bt$ と書ける。$c=ab-b-a$ のときは，そのようには書けない。

問題 3.28. n を正の整数とするとき，D_n で n の正の約数の全体を表す。n_1 と n_2 を互いに素な正の整数とすると，(d_1,d_2) を d_1d_2 に対応させる写像は，$D_{n_1}\times D_{n_2}$ と $D_{n_1n_2}$ の間の全単射である。

定義 3.29. 任意の素数 p に対して $\mathbb{Z}\to\mathbb{N}\cup\{\infty\}$ への写像 ν_p を次のように定義する。まず $\nu_p(0)=\infty$ である。$n\neq 0$ に対しては，$n=p^em$ かつ $p\perp m$ であるような非負の整数 e を $\nu_p(n)$ と定義する。

このとき $n \neq 0$ の素因数分解は

$$n = \mathrm{sgn}(n) \prod_p p^{\nu_p(n)}$$

という形に書ける。積は，すべての素数 p に亘ってとられるが，有限個の例外を除いて $\nu_p(n) = 0$ だから，実は有限個の素数ベキの積である。任意の $a, b \in \mathbb{Z}$ と任意の素数 p に対して $\nu_p(ab) = \nu_p(a) + \nu_p(b)$ である。また a が b の約数であるための必要十分条件は，任意の素数 p に対して $\nu_p(a) \leqq \nu_p(b)$ であることだ。ただし，a や b が 0 の場合にもうまく行くように，$n + \infty = \infty$，$n \leqq \infty$ と定義する。従って，

$$\gcd(a, b) = \prod_p p^{\min(\nu_p(a), \nu_p(b))}$$

である。

定義 3.30. $a, b \in \mathbb{Z}$ とする。a と b の共通の倍数を a と b の**公倍数**（common multiple）という。さらに m が非負で，他のすべての公倍数の約数なら**最大公倍数**（LCM, least common multiple）という。a と b の最大公約数は，ただ 1 つ存在する。それを $\mathrm{lcm}(a, b)$ と記す。

$a, b \in \mathbb{Z}$ の一方が 0 なら，公倍数は 0 だけだから，$\mathrm{lcm}(a, b) = 0$ となる。一般に，任意の $a, b \in \mathbb{Z}$ に対して

$$\mathrm{lcm}(a, b) = \prod_p p^{\max(\nu_p(a), \nu_p(b))}$$

である。

最大公約数，最小公倍数の概念は，任意複数個の整数に対して一般化できる $a_1, a_2, \ldots, a_k \in \mathbb{Z}$ とするとき，そのすべてに共通する約数を $a_1, a_2, \ldots, a_k$ の**公約数**という。さらに d が非負で，他のすべての公約数の倍数なら**最大公約数**という。$a_1, a_2, \ldots, a_k \in \mathbb{Z}$ の最大公約数は，ただ 1 つ存在し，それを $\gcd(a_1, a_2, \ldots, a_k)$ と記すが，

$$\gcd(a_1, a_2, \ldots, a_k) = \prod_p p^{\min(\nu_p(a_1), \nu_p(a_2), \ldots, \nu_p(a_k))}$$

である。対称的に $a_1, a_2, \ldots, a_k$ のすべてに共通する倍数を $a_1, a_2, \ldots, a_k$ の**公倍数**という。さらに m が非負で，他のすべての公倍数の約数なら**最小公倍数**という。$a_1, a_2, \ldots, a_k \in \mathbb{Z}$ の最大公倍数は，ただ 1 つ存在し，それを $\mathrm{lcm}(a_1, a_2, \ldots, a_k)$ と記すが，

$$\mathrm{lcm}(a_1, a_2, \ldots, a_k) = \prod_p p^{\max(\nu_p(a_1), \nu_p(a_2), \ldots, \nu_p(a_k))}$$

である。

$a_1, a_2, \ldots, a_k \in \mathbb{Z}$ がどの 2 つをとっても互いに素なとき，すなわち $i \neq j$ ならば $a_i \perp a_j$ のとき，**対ごとに互いに素**（pairwise relatively prime）という。

有理数 $r \in \mathbb{Q}$ は整数 $a, b \in \mathbb{Z}$ により，$r = a/b$ と書ける。特に $\gcd(a, b) = d$ とすると，$a_0 = a/d$，$b_0 = b/d$ も整数であり，$a_0 \perp b_0$ かつ $r = a/b = a_0/b_0$ である。このように，互いに素な整数を分子分母を用いて書き表した有理数を**既約分数**という。有理数の既約分数表現で，特に分母を正とするようなものは唯一に定まる。

問題 3.31. $a_1, a_2, \ldots, a_k \in \mathbb{Z}$ を対ごとに互いに素な整数とする。全 i について $a_i \backslash n$ のとき $\left(\prod_{i=1}^{k} a_i\right) \backslash n$ である。また，$\mathrm{lcm}(a_1, a_2, \ldots, a_k) = \prod_{i=1}^{k} a_i$ である。

問題 3.32. 任意の $a, b, c \in \mathbb{Z}$ に対して次が成立する。

(a) $\mathrm{lcm}(a, b) = \mathrm{lcm}(b, a)$
(b) $\mathrm{lcm}(a, b) = |a| \iff b \perp a$
(c) $\mathrm{lcm}(a, a) = \mathrm{lcm}(a, 1) = |a|$
(d) $\mathrm{lcm}(ca, cb) = |c| \,\mathrm{lcm}(a, b)$
(e) $\gcd(a, b) \,\mathrm{lcm}(a, b) = |ab|$
(f) $\gcd(a, b) = 1 \iff \mathrm{lcm}(a, b) = |ab|$
(g) $\gcd(a + b, \mathrm{lcm}(a, b)) = \gcd(a, b)$

問題 3.33. $a_1, a_2, \ldots, a_k \in \mathbb{Z}$ かつ $\gcd(a_1, a_2, \ldots, a_k) = d$ とする。このとき $d\mathbb{Z} = a_1\mathbb{Z} + a_2\mathbb{Z} + \cdots + a_k\mathbb{Z}$ である。特に整数 $z_1, z_2, \ldots, z_k \in \mathbb{Z}$ が存在して, $d = a_1 z_1 + a_2 z_2 + \cdots + a_k z_k$ と書ける。

問題 3.34. 任意の有理数 $x \in \mathbb{Q}$ は

$$x = \mathrm{sgn}(x) p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$$

の形に，$p_i^{e_i}$ の並び順を除いて一意に書ける。ここで p_i は互いに異なる素数で，e_i は 0 でない整数である。

問題 3.35. n と k を正の整数とし，$x \in \mathbb{R}$ を $x^k = n$ を満たす実数とする。このとき x は整数でなければ無理数である。

問題 3.36. 任意の正の整数 k に対して，連続する k 個以上の整数で，すべて合成数になるものが存在する。

問題 3.37. p を素数とする。任意の正の整数 n に対して

$$\nu_p(n!) = \sum_{k \geqq 1} \left\lfloor \frac{n}{p^k} \right\rfloor = \frac{n - \delta_p(n)}{p - 1}$$

を示せ。ここで $\delta_p(n)$ は，n を p 進表記したときの各桁の総和（いわゆる p 進数字根）で，例えば $\delta_2(25) = 1 + 1 + 0 + 0 + 1 = 3$，$\delta_3(25) = 2 + 2 + 1 = 5$，$\delta_5(25) = 1 + 0 + 0 = 1$ である。

問題 3.38. p を素数とする。任意の整数 $a, b \in \mathbb{Z}$ に対して $\nu_p(a + b) \geqq \min(\nu_p(a), \nu_p(b))$ であり，かつ $\nu_p(a) < \nu_p(b)$ ならば $\nu_p(a + b) = \nu_p(a)$ である。

問題 3.39. p を素数とする。関数 ν_p の定義域を $\mathbb{Z}$ から $\mathbb{Q}$ に拡張して，$a, b \in \mathbb{Z}$ に対し $\nu_p(a/b)$ を $\nu_p(a/b) \stackrel{\mathrm{def}}{=} \nu_p(a) - \nu(b)$ と定義する。

(a) 定義 ν_p は，well-defined である，すなわち有理数の分数表現によらない。
(b) 任意の $x, y \in \mathbb{Q}$ に対して $\nu_p(xy) = \nu_p(x) + \nu_p(y)$ である。
(c) 任意の $x, y \in \mathbb{Q}$ に対して $\nu_p(x+y) \geqq \min\{\nu_p(x), \nu_p(y)\}$ かつ $\nu_p(x) < \nu_p(y)$ ならば $\nu_p(x+y) = \nu_p(x)$ である。

(d) 0 でない有理数 $x \in \mathbb{Q}$ は $x = \prod_p p^{\nu_p(x)}$ と書ける。ここで，積は全素数に亘ってとるが，有限個の素数 p を除いて $\nu_p(x) = 0$ となるので，実質的には有限個の積である。
(e) $x \in \mathbb{Z}$ であるための必要十分条件は，任意の素数 p について $\nu_p(x) \geqq 0$ が成り立つことだ。

問題 3.40. n を 1 より大きい整数とするとき，$\sum_{i=1}^{n} 1/i$ は整数ではない。

3.2 合同関係 ☆

3.2.1 合同と法計算 ☆

定義 3.41. n を正の整数とする。$a, b \in \mathbb{Z}$ を整数とする。$n \setminus (a-b)$ のとき，n を**法** (modulo) として a と b は**合同** (congruent) といい，$a \equiv b \pmod{n}$ と記す。これは，$a \bmod n = b \bmod n$ であることに他ならない。

同様に a_1, a_2, b が一般の実数の場合にも，$(a_1 - a_2)/b \in \mathbb{Z}$ であることを $a_1 \equiv a_2 \pmod{b}$ と記す。

定理 3.42. n を正の整数とする。n を法として合同という関係は同値関係である。すなわち，任意の $a, b, c \in \mathbb{Z}$ について次が成り立つ。

(反射律) $a \equiv a \pmod{n}$
(対称律) $a \equiv b \pmod{n}$ ならば $b \equiv a \pmod{n}$
(推移律) $a \equiv b \pmod{n}$，$b \equiv c \pmod{n}$ ならば $a \equiv c \pmod{n}$

証明: 簡単な計算で示される。 □

定理 3.43. n を正の整数とする。$a, b, a', b' \in \mathbb{Z}$ が $a \equiv a' \pmod{n}$，$a \equiv a' \pmod{n}$ を満たせば $a + b \equiv a' + b' \pmod{n}$，$ab \equiv a'b' \pmod{n}$ である。
証明: 簡単な計算で示される。 □

定理 3.44. n を正の整数とする。$a \in \mathbb{Z}$ に対して，$z \equiv a \pmod{n}$ かつ $0 \leqq z < n$ なる $z \in \mathbb{Z}$ がただ 1 つ定まる。もっと一般には，任意の $x \in \mathbb{R}$ と任意の $a \in \mathbb{Z}$ に対して，$z \equiv a \pmod{n}$ かつ $z \in [x..x+n)$ なる $z \in \mathbb{Z}$ がただ 1 つ定まる。
証明: 剰余つき除算に関する定理 3.8 からの直接の帰結である。 □

問題 3.45. n を正の整数，$P(x) \in \mathbb{Z}[x]$ を x についての整係数多項式とする。$a \equiv b \pmod{n}$ ならば，$P(a) \equiv P(b) \pmod{n}$ である。

問題 3.46. n, n' を正の整数とする。$a \equiv b \pmod{n}$ かつ $a \equiv b \pmod{n'}$ であるための必要十分条件は $a \equiv b \pmod{\operatorname{lcm}(n, n')}$ である。

問題 3.47. n を正の整数とする。$a \equiv b \pmod{n}$ ならば，$\gcd(a, n) = \gcd(b, n)$ である。

法計算の一番簡単な応用例は数値計算の検算に用いることである。例えば，$123 \times 341 = 41843$ という計算が正しいかどうか，簡単に見当をつける方法に「九去法」というのがある。これは計算に用いた数値そのものの代わりに，数値の各桁の和を用いて検算する方法である。数値の各桁の和をその数値の（10 進）**数字根**と呼ぶ。上の例ならば 123 の数字根 $1+2+3=6$，341 の数字根 $3+4+1=8$，41843

の数字根 $4+1+8+4+3=20$ を代わりに用いて計算すると $6\times 8=48\not\equiv 20 \pmod 9$ であるから，上の計算には間違いがあることが分かる。これはある数値とその数字根は，9 で割った余りが必ず等しくなることを利用した方法である。数字根を取った結果が十分小さい数でなければ，再び数字根をとってやってもいいが，数字根を計算する途中で合計が 9 になる数字を無視して計算する方が簡単だろう。おそらく，これが九去法という名の由来である。例えば，$12345\times 34567=426739615$ の検算であれば，12345 の数字根は $1+2+3+4+5=15$ だが，$4+5=9$ なのでそれを無視すれば $12345\equiv 6 \pmod 9$ が暗算で分かる。同様に 34567 は，$4+5=3+6=9$ なので $34567\equiv 7 \pmod 9$ が直ちに分かる，一方 $426739615\equiv 7 \pmod 9$ も同様に容易に得られるが， $6\times 7=42\not\equiv 7 \pmod 9$ だから，計算に間違いがあることが分かる。

このような検算法は，正しい計算を間違いと判定することはないが，間違いがあっても確実にそれが分かるとは限らない。しかし，数字がランダムに選ばれたとしたとき，9 で割った余りが一致する確率は 1/9 であるから，間違いを見逃す可能性はそれほど大きくない。もし，不安ならば，11 で割った余りなども併用すれば，見逃す確率はさらに下がり 1/99 になる。ちなみに 11 で割った余りは (1 の位を正として) 各桁の数字を交互に符号を変えながら加えることで簡単に求まる。例えば，計算 $12345\times 34567=426792615$ の場合，九去法では間違いが検出されないが，$5-4+3-2+1=3$，$7-6+5-4+3=5$，$5-1+6-2+9-7+6-2+4=18$ より，

$$12345\times 34567\equiv 3\times 5\equiv 15\not\equiv 426792615\equiv 18 \pmod{11}$$

となり，間違いであることが分かる。一方，正しい計算 $12345\times 34567=426729615$ の場合，$5-1+6-9+2-7+6-2+4=4$ だから，当然ではあるがテストに合格する。少なくとも，計算は 98/99 の確率で正しいと言える。

問題 3.48. 正の整数とその数字根は，9 で割ったときの余りが等しいことを証明せよ。また，正の整数の各桁の数字を上のように交互に符号を変えながら加えた結果と元の整数とは，11 で割ったときの余りが等しいことを証明せよ。

9 や 11 に限らず，もっと別の数で割った余りを考えて行くことで，計算間違いの検出確率をさらに上げることができる。一般の数で割った余りの場合，数字根のようなものを利用する簡単な計算方法があるとは限らないが，それでも検算としては十分役に立つ。2.4.2 節で見たように，多精度整数を単精度整数で割った余りは高速に計算できる。例えばコンピュータによる多精度計算では，1000 桁掛ける 1000 桁が 2000 桁になるような計算をしばしばするが，その検算が単精度でできて，信頼度が大きいということは，きわめて有用である。

この手法は多項式の計算の場合にも利用できる。多項式 f,g,h に対して，計算 $f\cdot g=h$ を検算するには，勝手に選んだある値 a を多項式に代入して $f(a)\cdot g(a)=h(a)$ を確かめるということが行われるが，これは一次式 $x-a$ で多項式を割った余りの計算をしていることに他ならない。

類似の手法は他にも多くの応用を持つ[3]。

[3]法計算とは限らず，一般にデータ d にある関数 f (ハッシュ関数と呼ぶ) を適用した結果 $f(d)$ を d の代わりに用いて認証などを行う手法は，**指紋法**と呼ばれ，広く応用されている。$f(d)$ は d の**指紋** (フィンガープリント, fingerprint) と呼ばれるが，2 つの指紋が偶然に一致する可能性がほとんどないように f は選ばれる。上のように 2 つの巨大データの整合性のチェックなどにも有用であるが，例えば，パスワード認証などにも使われ，その場合，ハッシュ関数 f は，それを知っていても，指紋 $f(d)$ から元の d を求めることが困難であるように，普通は設計される。

問題 3.49. p を 2 以上の整数とする。a を正の整数とするとき，その p 進表記を $(a_k a_{k-1} \cdots a_1 a_0)_p$ と記す。すなわち，$a_i \in [0..p)$ で $a = \sum_{i=0}^{k} a_i p^i$ である。このとき各桁の数字の和 $b \stackrel{\text{def}}{=} \sum_{i=0}^{k} a_i$ を a の p **進数字根**と呼ぶ。$a \equiv b \pmod{p-1}$ である。

問題 3.50. e を正の整数とする。$a \in [0..2^e)$ に対して，a の e ビット 2 進表記を反転した列が表す数を $\tilde{a}$ とすると，$\tilde{a} + 1 \equiv -a \pmod{2^e}$ である。(これが，計算機内で負の整数を表現するときに 2 の補数を用いる理由である。)

問題 3.51. 方程式 $7y^3 + 2 = z^3$ は整数解 $y, z \in \mathbb{Z}$ を持たない。

3.2.2 線形合同式 ☆

n を正の整数とする。本節では，整数 a と b が与えられたときの方程式 $az \equiv b \pmod{n}$ の解について考える。

定理 3.52. n を正の整数とする。$b, a \in \mathbb{Z}$，$d = \gcd(a, n)$ とする。

(1) $\exists z \in \mathbb{Z}\ az \equiv b \pmod{n} \iff d \setminus b$
(2) $\forall z \in \mathbb{Z}\ (az \equiv 0 \pmod{n} \iff z \equiv 0 \pmod{n/d})$
(3) $\forall z, z' \in \mathbb{Z}\ (az \equiv az' \pmod{n} \iff z \equiv z' \pmod{n/d})$

証明: (1) $\exists z \in \mathbb{Z}\ az \equiv b \pmod{n} \iff \exists z, q \in \mathbb{Z}\ az - nq = b \iff \gcd(a, n) \setminus b$ (系 3.18)
(2) $a/d \perp n/d$ だから，$az \equiv 0 \pmod{n} \iff n \setminus az \iff n/d \setminus (a/d)z \iff n/d \setminus z$
(3) 上より $az \equiv az' \pmod{n} \iff a(z - z') \equiv 0 \pmod{n} \iff n/d \setminus (z - z') \iff z \equiv z' \pmod{n/d}$ □

定義 3.53. n を正の整数とする。$a \in \mathbb{Z}$ に対し，n を法とする a の**逆数** (inverse) とは，$az \equiv 1 \pmod{n}$ なる整数 $z \in \mathbb{Z}$ のことである。

定理 3.54. n を正の整数とする。整数 a が n を法とする逆数を持つための必要十分条件は $n \perp a$ であり，z と z' がともに n を法とする a の逆数ならば，$z \equiv z' \pmod{n}$ である。また，$a \equiv a' \pmod{n}$ で z が n を法とする a の逆数ならば，z は n を法とする a' の逆数でもある。
証明: 定理 3.52 の (1) と (3) による。 □

定義 3.55. z が n を法とする a の逆数であることを $a^{-1} \equiv z \pmod{n}$ と記す。$n \perp a$ のとき，n を法とする a の逆数 z のうち $z \in [0..n)$ であるものを $a^{-1} \bmod n$ と表し，$az = b \pmod{n}$ であるような $z \in [0..n)$ を $b/a \bmod n$ と表す。

さらに一般に $d = \gcd(n, a)$ で $d \setminus b$ のときを考える。この場合，$a' = a/d$，$b' = b/d$，$n' = n/d$ とすると，$a' \perp n'$ で

$$\forall z \in \mathbb{Z}\ (az \equiv b \pmod{n}) \iff \forall z \in \mathbb{Z}\ (a'z \equiv b' \pmod{n'})$$

である。従って左辺の解を z を求めることは右辺の解を求めることであり，$z \equiv b'/a' \bmod n'$ だ。よって $z_0 = b'/a' \bmod n'$ とすると，$[0..n)$ 内には左辺の解は d 個ある。(具体的には $z_0, z_0 + n', \ldots, z_0 + (d-1)n'$ の d 個である。)

問題 3.56. $a_1, a_2, \ldots, a_k, b \in \mathbb{Z}$ とするとき，方程式 $a_1x_1 + \cdots + a_kx_k = b$ を考える。特に未知数 x_i の値を整数に制限したものを**線形ディオファンタス方程式** (Linear Diophantine equation) と呼ぶ。線形ディオファンタス方程式 $a_1x_1 + \cdots + a_kx_k = b$ が解 $(x_1, \ldots, x_k) \in \mathbb{Z}$ を持つための必要十分条件は $\gcd\{a_1, \ldots, a_k\} | b$ である。

問題 3.57. p を素数とする。e が正の整数，a, b, c が整数で $a \not\equiv 0 \pmod{p^{e+1}}$，$0 \leqq c < p^e$ とする。このとき

$$\left\lfloor ((az + b) \bmod p^{2e})/p^e \right\rfloor = c$$

を満たす整数 $z \in [0..p^{2e})$ の個数は p^e である。

3.2.3 中国剰余定理 ☆

本節では連立線形合同方程式の解について考える。

定理 3.58 (中国剰余定理)**.** $n_1, n_2, \ldots, n_k$ を対ごとに互いに素な正の整数とする。$a_1, a_2, \ldots, a_k$ を任意の整数とするとき

$$a \equiv a_i \pmod{n_i} \quad (i = 1, 2, \ldots, k)$$

を満たす解 $a \in \mathbb{Z}$ が存在する。さらに，$a' \in \mathbb{Z}$ もそのような解であるための必要十分条件は $a \equiv a' \pmod{n_1 n_2 \cdots n_k}$ である。

証明:

$$e_j \equiv [i = j] \pmod{n_i}$$

なる e_i を構成しよう。$n_i^* \stackrel{\text{def}}{=} \prod_{j \neq i} n_j$ とおくと，仮定より $n_i \perp n_i^*$ である。従って n_i を法とする n_i^* の逆数 t_i が存在するので，$e_i = t_i n_i^*$ とおけばいい。さらに $a = a_1e_1 + a_2e_2 + \cdots + a_ke_k$ とおけば，a は求める解である。

a' が別の解とだすると, 各 i について $a \equiv a_i \equiv a' \pmod{n_i}$ だから $n_i \setminus (a - a')$ である。n_i たちは，対ごとに互いに素だから $n_1 n_2 \cdots n_k \setminus (a - a')$ である。

逆に $a \equiv a' \pmod{n_1 n_2 \cdots n_k}$ ならば，明らかにどの i についても $a \equiv a' \pmod{n_i}$ である。従って a が解だから a' も解だ。 □

上の定理をもっと具体的に言えば，

$$\tau(a) = \langle a \bmod n_1, a \bmod n_2, \ldots, a \bmod n_k \rangle$$

で定義される $[0..n_1 n_2 \cdots n_k)$ から $[0..n_1) \times [0..n_2) \times \cdots \times [0..n_k)$ への関数 τ が全単射だということである。

問題 3.59. $a \equiv 2 \pmod 5$，$a \equiv 3 \pmod 6$，$a \equiv 5 \pmod 7$ を満たす整数 a を求めよ。

問題 3.60. $a \equiv 1 \pmod 3$，$a \equiv -1 \pmod 5$，$a \equiv 5 \pmod 7$ を満たす整数 a を求めよ。

問題 3.61. $n_1, \ldots, n_k$ を対ごとに互いに素な正の整数とする。$a_1, \ldots, a_k$ と $b_1, \ldots, b_k$ とを整数とする。$a_1 z \equiv b_1 \pmod{n_1}, \ldots, a_k z \equiv b_k \pmod{n_k}$ なる整数 z が存在するための必要十分条件は，$\gcd(a_1, n_1) \setminus b_1, \ldots, \gcd(a_k, n_k) \setminus b_k$ である。

問題 3.62. ν_p を定義 3.29 で定義したものとする。$p_1, \ldots, p_k$ を異なる素数，$a_1, \ldots, a_k$ を任意の整数，$e_1, \ldots, e_k$ を任意の非負整数とする。$\nu_{p_1}(a - a_1) = e_1, \ldots, \nu_{p_k}(a - a_k) = e_k$ なる整数 a が存在する。

問題 3.63. n_1 と n_2 を正の整数とする。a_1 と a_2 を任意の整数とすると，$a \equiv a_1 \pmod{n_1}$ かつ $a \equiv a_2 \pmod{n_2}$ なる整数 a が 存在するための必要十分条件は $a_1 \equiv a_2 \pmod{\gcd(n_1, n_2)}$ である。

3.3 剰余類

本節以降では，例えば，環，群，イデアルなど，基本的な代数の用語や記法を用いることがある。読者は必要に応じて適当な代数の本を参照されたい。

3.3.1 剰余類

定義 3.64. n を法とする合同関係 $\cdot \equiv \cdot \pmod{n}$ が作る同値類を $[\ ]_n$ （または n が明白であれば，単に $[\ ]$）と記し，n を法とする**剰余類**（residue class） と呼ぶ。すなわち

$$z \in [a]_n \iff z \equiv a \pmod{n} \iff \exists y \in \mathbb{Z}\ z = a + ny$$

である。n を法とする剰余類の全体を $\mathbb{Z}/n\mathbb{Z}$ と記す。すなわち $\mathbb{Z}/n\mathbb{Z} \stackrel{\text{def}}{=} \{[a]_n \mid a \in \mathbb{Z}\}$ である。

次のように加法と乗法を定義することで $\mathbb{Z}/n\mathbb{Z}$ は可換環になる。

$$[a]_n + [b]_n \stackrel{\text{def}}{=} [a + b]_n, \quad [a]_n \cdot [b]_n \stackrel{\text{def}}{=} [a \cdot b]_n$$

慣例により乗法記号は $\cdot$ は省略されることが多く，$[a]_n \cdot [b]_n$ は単に $[a]_n[b]_n$ と書かれる。また，文脈から明らかならば，同値類の記号 $[\ ]_n$ や $[\ ]$ も省略され，代表元だけで表示されることがある。

定理 3.65. 上の加法，乗法の定義は well-defined である。$[0]_n$ は加法の単位元であり，$[-a]_n$ は $[a]_n$ の加法の逆元である。$[1]_n$ は乗法の単位元である。$[b]_n$ が $[a]_n$ の乗法逆元 ${[a]_n}^{-1}$ であるための必要十分条件は b が n を法とする a の逆数だということであり，$[a]_n$ が乗法逆元を持つための必要十分条件は $a \perp n$ である。

証明: いずれも定義と定理 3.43，3.54 からの明らかな帰結である。 □

定義 3.66. 乗法単位元を持つ任意の可換環 R において，乗法逆元を持つ要素を**単元**（unit）呼ぶ。R の単元の全体を $R^\times$ と記す。$R^\times$ は乗法に関して常に群をなす。（定理 3.65 から分かるように，$R^\times = (\mathbb{Z}/n\mathbb{Z})^\times = \{[a]_n \mid a \perp n\}$ である。p が素数ならば，明らかに $(\mathbb{Z}/p\mathbb{Z})^\times = (\mathbb{Z}/p\mathbb{Z}) \setminus \{[0]_p\}$ である。）

一般に乗法の単位元 1 を持つ可換環 R の要素 $r \in R$ と正の整数 $n \in \mathbb{Z}$ について

$$0 \cdot r \stackrel{\text{def}}{=} 0 \in R, \quad n \cdot r \stackrel{\text{def}}{=} \underbrace{r + \cdots + r}_{n}, \quad (-n) \cdot r \stackrel{\text{def}}{=} -(n \cdot r)$$

$$r^0 \stackrel{\text{def}}{=} 1 \in R, \quad r^n \stackrel{\text{def}}{=} \underbrace{r \cdots r}_{n}, \quad r^{-n} \stackrel{\text{def}}{=} (r^{-1})^n \ (r \in R^\times \text{ のとき})$$

と定義する。

問題 3.67. n を正の整数，$P(x) \in \mathbb{Z}[x]$ を x についての整係数多項式とすると，任意の整数 $a \in \mathbb{Z}$ について $P([a]_n) = [P(a)]_n$ である。

次は定理 3.58（中国剰余定理）の言い換えである。

定理 3.68 (中国剰余写像)**.** $n_1, n_2, \ldots, n_k$ を対ごとに互いに素な正の整数とし，$n = n_1 \cdots n_k$ とする。写像 $\theta : \mathbb{Z}/n\mathbb{Z} \to (\mathbb{Z}/n_1\mathbb{Z}) \times \cdots \times (\mathbb{Z}/n_k\mathbb{Z})$ を

$$\theta([a]_n) = \langle [a]_{n_1}, \ldots, [a]_{n_k} \rangle$$

で定義すると，θ の定義は well-defined であり，θ は環同型である。すなわち θ は全単射であり，任意の $\alpha, \beta \in \mathbb{Z}/n\mathbb{Z}$ に対して，$\theta(\alpha) = \langle \alpha_1, \ldots, \alpha_k \rangle$，$\theta(\beta) = \langle \beta_1, \ldots, \beta_k \rangle$ とおけば，

(a) $\theta(\alpha + \beta) = \langle \alpha_1 + \beta_1, \ldots, \alpha_k + \beta_k \rangle$
(b) $\theta(-\alpha) = \langle -\alpha_1, \ldots, -\alpha_k \rangle$,
(c) $\theta(\alpha\beta) = \langle \alpha_1\beta_1, \ldots, \alpha_k\beta_k \rangle$
(d) $\alpha \in (\mathbb{Z}/n\mathbb{Z})^\times$ であるための必要十分条件は $\alpha_1 \in (\mathbb{Z}/n_1\mathbb{Z})^\times, \ldots, \alpha_k \in (\mathbb{Z}/n_k\mathbb{Z})^\times$ であり，その場合 $\theta(\alpha^{-1}) = \langle \alpha_1^{-1}, \ldots, \alpha_k^{-1} \rangle$ である。

証明: θ が well-defined であることは，$a \equiv a' \pmod{n}$ からすべての i について $a \equiv a' \pmod{n_i}$ であることが導かれるので明らかである。θ が全単射であることは，定理 3.58（中国剰余定理）の内容そのものである。(a)–(d) も $\alpha = [a]_n$，$\beta = [b]_n$ などとして，代表元 a，b に置き換えて考えれば簡単な計算で示される。例えば (d) の場合，まず，

$$\alpha \in (\mathbb{Z}/n\mathbb{Z})^\times \iff a \perp n \iff \forall i \in \{1, \ldots, k\}\ a \perp n_i \iff \forall i \in \{1, \ldots, k\}\ \alpha_i \in (\mathbb{Z}/n_i\mathbb{Z})^\times$$

である。また，$\beta = \alpha^{-1}$ とすれば，

$$(\alpha_1\beta_1, \ldots, \alpha_k\beta_k) = \theta(\alpha\beta) = \theta([1]_n) = ([1]_{n_1}, \ldots, [1]_{n_k})$$

だから，どの i についても $\alpha_i\beta_i = [1]_{n_i}$ すなわち $\alpha_i^{-1} = \beta_i$ であることが示される。 □

問題 3.69. θ を上の中国剰余写像とし，$P(x) \in \mathbb{Z}[x]$ を x についての整係数多項式とする。任意の $\alpha \in \mathbb{Z}/n\mathbb{Z}$ に対し，$\theta(\alpha) = \langle \alpha_1, \ldots, \alpha_k \rangle$ とおけば，$\theta(P(\alpha)) = \langle P(\alpha_1), \ldots, P(\alpha_k) \rangle$ である。

問題 3.70. p を奇素数とする。$\sum_{\beta \in (\mathbb{Z}/p\mathbb{Z})^\times} \beta = \sum_{\beta \in (\mathbb{Z}/p\mathbb{Z})^\times} (-\beta) = \sum_{\beta \in (\mathbb{Z}/p\mathbb{Z})^\times} \beta^{-1} = 0$ である。

問題 3.71. p を奇素数とする。$\displaystyle\sum_{i=1}^{p-1} \frac{1}{i}$ を既約分数として表現したとき，その分子は p で割り切れる。

問題 3.72. n を無平方（問題 3.25 参照）とし，$\alpha, \beta, \gamma \in \mathbb{Z}/n\mathbb{Z}$ とする。$\alpha^2\beta = \alpha^2\gamma$ ならば $\alpha\beta = \alpha\gamma$ である。

3.3.2 オイラーの φ 関数 ☆

定義 3.73. 正の整数 n に対して $(\mathbb{Z}/n\mathbb{Z})^\times$ の要素数，すなわち

$$\varphi(n) = \#(\mathbb{Z}/n\mathbb{Z})^\times$$

として，オイラーの関数 φ は定義される。

定理 3.74. $n_1, n_2, \ldots, n_k$ を対ごとに互いに素な正の整数とする。

$$\varphi(n_1 n_2 \cdots n_k) = \varphi(n_1)\varphi(n_2)\cdots\varphi(n_k)$$

である。

証明: 定理 3.68 の中国剰余写像 θ を考える。定理の (d) より，θ は $(\mathbb{Z}/n\mathbb{Z})^\times$ と $(\mathbb{Z}/n_1\mathbb{Z})^\times \times \cdots \times (\mathbb{Z}/n_k\mathbb{Z})^\times$ との間の全単射を与える。よって φ の定義より上の関係は明らかである。 □

定理 3.75. p を素数とし，e を正の整数とすると，次が成り立つ。

$$\varphi(p^e) = p^{e-1}(p-1) = p^e\left(1 - \frac{1}{p}\right)$$

証明: $\varphi(p^e)$ は p^e より小さくて p^e と互いに素な非負整数の数である。p^e と互いに素でないのは p の倍数であるから，p^e 未満の非負整数には $0, p, 2p, \ldots, p^e - 2p, p^e - p$ がある。これらは全部で p^{e-1} 個だから，$\varphi(p^e) = p^e - p^{e-1} = p^{e-1}(p-1)$ である。 □

定理 3.76. n の素因数分解を $n = p_1^{e_1}\cdots p_r^{e_r}$ とする，次が成り立つ。

$$\varphi(n) = \prod_{i=1}^{r} p^{e_i-1}(p_i - 1) = n\prod_{i=1}^{r}\left(1 - \frac{1}{p_i}\right)$$

証明: 前 2 定理より明らか。 □

問題 3.77. 任意の正の整数 $n, m \in \mathbb{Z}$ に対して $\varphi(nm) = \gcd(n, m) \cdot \varphi(\mathrm{lcm}(n, m))$ である。

問題 3.78. n が r 個の異なる奇素数で割り切れるなら，$2^r \setminus \varphi(n)$ である。

3.3.3 オイラーの定理とフェルマーの小定理 ☆

定義 3.79. 一般に，乗法単位元 1 を持つ有限可換環 R を考える。$r \in R$ が零因子でないとき，その冪の列を $r^0(=1), r^1(=r), r^2, \ldots, r^k, \ldots$ とすると，R は有限集合だから，この中には同じものが存在する。そのような $r^i = r^j$ $(i < j)$ をとると r は零因子ではないので，r^i を相殺して $1 = r^0 = r^{j-i}$ となる。すなわち，$r^k = 1$ となる k が存在するので，そのような最小の正の整数を r の**乗法位数**（multiplicative order）といい，$\mathrm{ord}(r)$ と記す。n を正の整数とし，特に $R = \mathbb{Z}/n\mathbb{Z}$ のとき，$[a]_n$ の乗法位数を，整数 a の（n を法とする）**乗法位数**と呼び，単に $\mathrm{ord}(a)$ と記す。すなわち整数 a の n を法とする乗法位数 $\mathrm{ord}(a)$ とは，$a^k \equiv 1 \pmod{n}$ となる最小の正の整数 k のことである。

定理 3.80. r を可換環 R の元，$k = \mathrm{ord}(r)$ とする。$i \in \mathbb{Z}$ を任意の整数とするとき，$r^i = 1$ となるための必要十分条件は $k \setminus i$ である。もっと一般には，任意の $i, j \in \mathbb{Z}$ について $r^i = r^j$ となるための必要十分条件は $i \equiv j \pmod{k}$ である。

定理 3.81 (オイラーの定理)**.** n を正の整数とする。任意の $\alpha \in (\mathbb{Z}/n\mathbb{Z})^\times$ に対して，$\alpha^{\varphi(n)} = 1$ である。特に $a \perp n$ なる任意の $a \in \mathbb{Z}$ について $\mathrm{ord}(a) \setminus \varphi(n)$ である。

定理 3.82 (フェルマーの小定理)**.** p を素数とすると，任意の $\alpha \in (\mathbb{Z}/p\mathbb{Z})$ に対して，$\alpha^p = \alpha$ である。すなわち，任意の整数 $a \in \mathbb{Z}$ に対して $a^p \equiv a \pmod{p}$ である。

問題 3.83. 定理 3.80，3.81，3.82 を証明せよ。（ヒント：定理 3.81 の証明は，いろいろ考えられるが，群論のラグランジュの定理を利用するのが一番簡潔だろう。）

問題 3.84. 21 個の整数があり，そのどれも 7 で割り切れないとする。それら 21 個の整数の 18 乗の和は 7 で割り切れることを証明せよ。

定義 3.85. n を正の整数とする。$a \perp n$ なる整数 $a \in \mathbb{Z}$ の乗法位数が $\varphi(n)$ のとき，a を法 n に対する**原始根**（primitive root）と呼ぶ。このとき $(\mathbb{Z}/n\mathbb{Z})^\times = \{[a^i]_n \mid i \in \mathbb{Z}\}$ と書ける。

法 n が原始根を持つためには，$n = 1, 2, 4$ であるか，素数 p と正の整数 e が存在して $n = p^e, 2p^e$ であることが必要十分であることを後で示す。

定理 3.86. r を可換環 R の元，k を r の乗法位数とする。$i \in \mathbb{Z}$ を任意の整数とするとき，r^i の乗法位数は $k/\gcd(i,k)$ である。
証明: 定義より r^i の乗法位数は $r^{im} = 1$ なる最小の正の整数 m である。

$$r^{im} = 1 \iff im \equiv 0 \pmod{k} \iff m \equiv 0 \pmod{k/\gcd(i,k)} \text{（定理 3.52 (2)）}$$

□

問題 3.87.
法 19 に対する原始根をすべて見つけよ。

問題 3.88. n を 2 以上の整数とする。n が素数であるための必要十分条件は，n で割り切れない任意の整数 $a \in \mathbb{Z}$ に対して $a^{n-1} \equiv 1 \pmod{n}$ が成り立つことである。

問題 3.89. p, q を異なる素数とする。任意の $\alpha \in (\mathbb{Z}/pq\mathbb{Z})^\times$ に対して $\alpha^{\mathrm{lcm}(p-1,q-1)} = 1$ である。

問題 3.90. n を 2 や 5 を約数に持たない正の整数とする。$n \setminus (10^e - 1)$ となる正の整数 e が存在する。

問題 3.91. n を 2 以上の整数とすると，n は $2^n - 1$ を割り切らない。

問題 3.92 (フェルマーの小定理の一般化)**.** 任意の正の整数 n と $\alpha \in \mathbb{Z}/n\mathbb{Z}$ について，$\alpha^n = \alpha^{n-\varphi(n)}$ である。

問題 3.93. 任意の素数 p と任意の整数 a に対して $(a+1)^p \equiv a^p + 1 \pmod{p}$ を示せ。また，そのことを用いてフェルマーの小定理を証明せよ。

3.4 平方剰余

R を単位元 1 を持つ環とする。その単元の m 乗の集合 $(R^\times)^m$ を $\{\beta^m \mid \beta \in R^\times\}$ で定義する。$1^m = 1$ だから，明らかに $1 \in (R^\times)^m$ である。

定理 3.94. R を可換環とし，m を正の整数とすると $(R^\times)^m$ は乗法に関して $R^\times$ の部分群である。すなわち $1 \in (R^\times)^m$ であり，$\alpha, \beta \in (R^\times)^m$ ならば $\alpha^{-1}\beta \in (R^\times)^m$ である。

問題 3.95. 上の定理 3.94 を証明せよ。

定理 3.96. R を可換環とし，$\alpha \in R^\times$ とする。l，m を正の整数とする。$l \perp m$ かつ $\alpha^l \in (R^\times)^m$ ならば $\alpha \in (R^\times)^m$ である。特に l が奇数で $\alpha^l \in (R^\times)^2$ ならば $\alpha \in (R^\times)^2$ である。
証明: $\alpha^l = \beta^m$ とする。$l \perp m$ だから，定理 3.17 より $sl + tm = 1$ なる整数 $s, t \in \mathbb{Z}$ が存在する。

$$\alpha = \alpha^{sl+tm} = \alpha^{sl}\alpha^{tm} = \beta^{sm}\alpha^{tm} = (\beta^s\alpha^t)^m \in (R^\times)^m$$

である。 □

以下では，主に正の整数 n と整数 a について，$[a]_n \in ((\mathbb{Z}/n\mathbb{Z})^\times)^2$ となる条件について考察する。

定義 3.97. n を正の整数とする。整数 a は，$a \perp n$ で $a \equiv b^2 \pmod{n}$ なる整数 b が存在するとき，すなわち $[a]_n \in ((\mathbb{Z}/n\mathbb{Z})^\times)^2$ のとき，(n を法とする）**平方剰余**（quadratic residue）と呼ばれる。このとき b を（n を法とする）a の**平方根**（square root）と呼ぶ。

3.4.1 奇素数を法とする平方剰余

まず奇素数 p を法とする平方剰余について考える。

定理 3.98. p を奇素数とし，$\alpha, \beta \in (\mathbb{Z}/p\mathbb{Z})^\times$ とする。$\alpha^2 = \beta^2$ であるための必要十分条件は $\alpha = \pm\beta$ である。特に $\alpha^2 = 1$ であるための必要十分条件は $\alpha = \pm 1$ である。（注: この定理は $p = 2$ のときにも成り立つが，その場合は $1 = -1$ であることに注意されたい）
証明: $\alpha = \pm\beta$ ならば，$\alpha^2 = \beta^2$ であることは明らか。逆に $\alpha^2 = \beta^2$ とする。$\alpha = [a]_p$，$\beta = [b]_p$ とおけば，$b^2 \equiv a^2 \pmod{p}$ だから，$b^2 - a^2 = (b+a)(b-a) \equiv 0 \pmod{p}$ である。p は素数だから，$b+a$ か $b-a$ は p で割り切れる。前者なら $\alpha = -\beta$，後者なら $\alpha = \beta$ である。 □

定理 3.99. p を奇素数とする。$\#((\mathbb{Z}/p\mathbb{Z})^\times)^2 = \dfrac{p-1}{2}$ である。
証明: $(\mathbb{Z}/p\mathbb{Z})^\times$ 上の関数 $f : x \mapsto x^2$ を考える。p が奇素数だから，$x \neq 0$ ならば $x \neq -x$ であり，従って定理 3.98 より，f は 2 対 1 の関数である。 □

定理 3.100 (ウィルソンの定理)**.** p が奇素数ならば，$\prod_{\beta \in (\mathbb{Z}/p\mathbb{Z})^\times} \beta = -1$ である。一般に正の整数 n が素数であるための必要十分条件は $(n-1)! \equiv -1 \pmod{n}$ である。
証明: p を奇素数としよう。$(\mathbb{Z}/p\mathbb{Z})^\times$ から $\kappa\lambda = 1$ となるような 2 点集合 $\{\kappa, \lambda\}$ を選び出していくことを考えると κ が決まれば λ は $1/\kappa$ として一意に定まる。また，定理 3.98 より $\beta^2 = 1$ となるような β は ± 1 であり，ちょうど 2 つ存在するので，$(\mathbb{Z}/p\mathbb{Z})^\times$ はちょうど $(p-3)/2$ 個の上のような 2 点集合と $\{1\}$ と $\{-1\}$ とに分割される。ゆえに $\prod_{\beta \in (\mathbb{Z}/p\mathbb{Z})^\times} \beta = 1^{(p-3)/2} \cdot 1 \cdot (-1) = -1$ である。これは $(p-1)! \equiv -1 \pmod{p}$ ということに他ならない。
逆に正の整数 n が $n_1 n_2$ と 2 つの 2 以上の整数の積に書けるなら，$(n-1)!$ は n_1 の倍数になるので $(n-1)! \equiv -1 \pmod{n}$ となることはない。 □

定理 3.101 (オイラーの基準)**.** p を奇素数とし，$\alpha \in (\mathbb{Z}/p\mathbb{Z})^\times$ とする。

(1) $\alpha^{(p-1)/2} = \pm 1$
(2) $\alpha \in ((\mathbb{Z}/p\mathbb{Z})^\times)^2 \Longrightarrow \alpha^{(p-1)/2} = 1$
(3) $\alpha \notin ((\mathbb{Z}/p\mathbb{Z})^\times)^2 \Longrightarrow \alpha^{(p-1)/2} = -1$

証明: $\gamma = \alpha^{(p-1)/2}$ とおく。

(1) については，オイラーの定理（定理 3.81）より $\gamma^2 = \alpha^{p-1} = 1$ だから，定理 3.98 より $\gamma = \pm 1$ である。

(2) を示すために $\alpha = \beta^2$ と仮定する。すると，オイラーの定理より $\alpha^{(p-1)/2} = \beta^{p-1} = 1$ である。

(3) を示すために $\alpha \notin ((\mathbb{Z}/p\mathbb{Z})^\times)^2$ と仮定する。ウィルソンの定理（定理 3.100）の積 $\prod_{\beta\in(\mathbb{Z}/p\mathbb{Z})^\times} \beta = -1$ を考える。$(\mathbb{Z}/p\mathbb{Z})^\times$ から $\kappa\lambda = \alpha$ となるような 2 点集合 $\{\kappa,\lambda\}$ を選び出していくことを考えると κ が決まれば λ は α/κ として一意に定まる。また，仮定より $\beta^2 = \alpha$ となるような β は存在しないので，$(\mathbb{Z}/p\mathbb{Z})^\times$ はちょうど $(p-1)/2$ 個の上のような 2 点集合に分割される。従って，$-1 = \prod_{\beta\in(\mathbb{Z}/p\mathbb{Z})^\times} \beta = \alpha^{(p-1)/2}$ である。 □

系 3.102. p を奇素数とし，$\alpha, \beta \in (\mathbb{Z}/p\mathbb{Z})^\times$ とする。$\alpha, \beta \notin ((\mathbb{Z}/p\mathbb{Z})^\times)^2$ ならば $\alpha\beta \in ((\mathbb{Z}/p\mathbb{Z})^\times)^2$ である。

3.4.2 奇素数のベキを法とする平方剰余

次に奇素数の冪 p^e を法とする平方剰余について考える。まず，定理 3.98 を一般化しよう。

定理 3.103. p を奇素数とし，e を正の整数，$\alpha, \beta \in (\mathbb{Z}/p^e\mathbb{Z})^\times$ とする。$\alpha^2 = \beta^2$ であるための必要十分条件は $\alpha = \pm\beta$ である。特に，$\alpha^2 = 1$ であるための必要十分条件は $\alpha = \pm 1$ である。（注: この定理は，定理 3.98 と異なり，$p = 2$ のときには成り立たないことに注意されたい）

証明: $\alpha = \pm\beta$ ならば，$\alpha^2 = \beta^2$ であることは明らか。逆に $\alpha^2 = \beta^2$ とする。$[a]_{p^e} = \alpha$，$[b]_{p^e} = \beta$ とおけば，$a^2 \equiv b^2 \pmod{p^e}$ だから，$a^2 - b^2 = (a+b)(a-b) \equiv 0 \pmod{p^e}$ である。$p \setminus (a+b)$ かつ $p \setminus (a-b)$ ならば，$a + b = jp$，$a - b = kp$ とおくことで，$2a = (j+k)p$，$2b = (j-k)p$ が得られるが，p が奇数だから，$p \setminus a$，$p \setminus b$ となり，$\alpha, \beta \in (\mathbb{Z}/p^e\mathbb{Z})^\times$ という条件に反する。従って，$a+b$ と $a-b$ は，一方が p と互いに素であり，他方が p^e で割り切れる。どちらが割り切れるかによって，$\alpha = -\beta$ または $\alpha = \beta$ が成り立つ。 □

定理 3.104. p を奇素数とし，e を正の整数とする。$\#((\mathbb{Z}/p\mathbb{Z})^\times)^2 = \dfrac{\varphi(p^e)}{2} = p^{e-1}\dfrac{p-1}{2}$ である。

証明: 定理 3.99 の証明と同様。 □

問題 3.105. p を奇素数，e を正の整数とするとき，$(\mathbb{Z}/p^e\mathbb{Z})^\times$ について，ウィルソンの定理（定理 3.100），オイラーの基準（定理 3.101）を拡張したものを述べ，証明せよ。

系 3.106. p を奇素数，e を正の整数とし，$\alpha, \beta \in (\mathbb{Z}/p^e\mathbb{Z})^\times$ とする。$\alpha, \beta \notin ((\mathbb{Z}/p^e\mathbb{Z})^\times)^2$ ならば $\alpha\beta \in ((\mathbb{Z}/p^e\mathbb{Z})^\times)^2$ である。

定理 3.107. p を奇素数，e を正の整数とする。a を任意の整数とすると，a が p^e を法とする平方剰余であるための必要十分条件は a が p を法とする平方剰余であることだ。

証明: ある整数 $b \in \mathbb{Z}$ に対して $a \equiv b^2 \pmod{p^e}$ とすれば，明らかに $a \equiv b^2 \pmod{p}$ でもある。

逆に a が p^e を法とする平方剰余でないとする。a が p で割り切れれば定義より a は p を法とする平方剰余でないので，$a \perp p$ と仮定する。問題 3.105 より $a^{p^{e-1}(p-1)/2} \equiv -1 \pmod{p^e}$ だから，$a^{p^{e-1}(p-1)/2} \equiv -1 \pmod{p}$ でもある。フェルマーの小定理（定理 3.82）を繰り返し用いて

$$a \equiv a^p \equiv \cdots \equiv a^{p^{e-1}} \pmod{p}$$

だから，

$$-1 \equiv a^{p^{e-1}(p-1)/2} \equiv a^{(p-1)/2} \pmod{p}$$

である。よってオイラーの基準より a は p を法とする平方剰余でない。 □

3.4.3 2^e を法とする平方剰余

明らかに，$(\mathbb{Z}/2\mathbb{Z})^\times = \{1\}$，$(\mathbb{Z}/4\mathbb{Z})^\times = \{1, -1\}$ だから，$((\mathbb{Z}/2\mathbb{Z})^\times)^2 = \{1\}$，$((\mathbb{Z}/4\mathbb{Z})^\times)^2 = \{1\}$ である。$e \geqq 3$ のとき，次が成り立つ。

補題 3.108. $e \geqq 3$ とするとき，方程式 $x^2 = 1$ の解は，$(\mathbb{Z}/2^e\mathbb{Z})^\times$ 内に 4 つあり，$x = \pm 1$ と $x = [2^{e-1} \pm 1]$ である。

証明: これらが実際に解であることは

$$(\pm 1)^2 = 1, \qquad [2^{e-1} \pm 1]^2 = [2^{2e-2} \pm 2 \cdot 2^{e-1} + 1] = 1$$

より分かる。逆に $x^2 = 1$ ならば，$x = [a]_{2^e}$ とおく。a はもちろん奇数だから，$a = 2m + 1$ $(m \in \mathbb{Z})$ とおくと，$a^2 - 1 = 4m(m+1) \equiv 0 \pmod{2^e}$ より，$m(m+1) \equiv 0 \pmod{2^{e-2}}$ である。m, $m+1$ の一方は奇数で $e \geqq 3$ だから，$m \equiv 0 \pmod{2^{e-2}}$ か $m + 1 \equiv 0 \pmod{2^{e-2}}$ である。それぞれに応じて $a = 2m + 1 \equiv 1 \pmod{2^{e-1}}$ か $a = 2m + 1 \equiv -1 \pmod{2^{e-1}}$ である。 □

定理 3.109. $e \geqq 3$ とし，$\alpha, \beta \in (\mathbb{Z}/2^e\mathbb{Z})^\times$ とする。$\alpha^2 = \beta^2$ であるための必要十分条件は，$\alpha = \pm\beta$ または $\alpha = [2^{e-1} \pm 1]\beta$ である。

証明: $\alpha^2 = \beta^2$ の両辺を β^2 で割れば，$(\alpha/\beta)^2 = 1$ である。すると補題 3.108 より，$\alpha/\beta = \pm 1$ または $\alpha/\beta = [2^{e-1} \pm 1]$ である。 □

定理 3.110. $e \geqq 3$ とするとき，$\#((\mathbb{Z}/2^e\mathbb{Z})^\times)^2 = \varphi(2^e)/4 = 2^{e-3}$ である。

証明: $((\mathbb{Z}/2^e\mathbb{Z})^\times)^2$ 上の関数 $f : x \mapsto x^2$ を考えると，上の定理より f は 4 対 1 の関数である。 □

系 3.111. $e \geqq 3$ とし，a を奇数とする。a が 2^e を法とする平方剰余であるための必要十分条件は $a \equiv 1 \pmod{8}$ である。

証明: $a = (2m+1)^2 = 4m(m+1) + 1$ とすれば，明らかに $a \equiv 1 \pmod{8}$ である。逆に，$(\mathbb{Z}/2^e\mathbb{Z})^\times$ 内には，このような数 a による剰余類 $[a]$ がちょうど 2^{e-3} 個しかないので，定理 3.110 より，このすべてが平方剰余である。 □

3.4.4 一般の正整数を法とする平方剰余

さて，一般の正の整数 n を法とする平方剰余について考えよう。

定理 3.112. $n = p_1^{e_1} \cdots p_r^{e_r}$ と素因数分解されるとし，$\theta : \mathbb{Z}/n\mathbb{Z} \to (\mathbb{Z}/p_1^{e_1}\mathbb{Z}) \times \cdots \times (\mathbb{Z}/p_r^{e_r}\mathbb{Z})$ を中国剰余写像とする。$\alpha \in (\mathbb{Z}/n\mathbb{Z})^\times$ が平方剰余であるための必要十分条件は，$\theta(\alpha) = \langle \alpha_1, \ldots, \alpha_r \rangle$ とするとき，各 α_i が $(\mathbb{Z}/p_i^{e_i}\mathbb{Z})^\times$ において平方剰余であることである。

証明: $\alpha = \beta^2$ と仮定し，$\theta(\beta) = \langle \beta_1, \ldots, \beta_r \rangle$ とおくと，

$$\langle \alpha_1, \ldots, \alpha_r \rangle = \theta(\alpha) = \theta(\beta^2) = \langle \beta_1^2, \ldots, \beta_r^2 \rangle$$

である。逆に各 i について $\alpha_i = \beta_i^2$ と書けるなら $\beta = \theta^{-1}\langle \beta_1, \ldots, \beta_r \rangle$ とおくことで，$\alpha = \beta^2$ となる。 □

特に θ は $((\mathbb{Z}/n\mathbb{Z})^\times)^2$ に制限することで $((\mathbb{Z}/p_1^{e_1}\mathbb{Z})^\times)^2 \times \cdots \times ((\mathbb{Z}/p_r^{e_r}\mathbb{Z})^\times)^2$ との間の全単射になる。それゆえ n が奇数なら，p_i もすべて奇数であり，

$$\#((\mathbb{Z}/n\mathbb{Z})^\times)^2 = \prod_{i=1}^{r} \frac{\varphi(p_i^{e_i})}{2} = \frac{\varphi(n)}{2^r}$$

である。$\alpha = \beta^2$ のとき，$\theta(\beta) = \langle \beta_1, \ldots, \beta_r \rangle$ とおくと，$\theta^{-1}\langle \pm\beta_1, \ldots, \pm\beta_r \rangle$ はどれも方程式 $x^2 = \alpha$ の解 x となるので，$((\mathbb{Z}/n\mathbb{Z})^\times)^2$ の要素 α は，一般に 2^r 個の平方根を持つ。

$n = 2p_1^{e_1} \cdots p_r^{e_r}$ （p_i は奇素数）と素因数分解される場合も n が奇数の場合と同様である。すなわち，$\#((\mathbb{Z}/n\mathbb{Z})^\times)^2 = \dfrac{\varphi(n)}{2^r}$ であり，$((\mathbb{Z}/n\mathbb{Z})^\times)^2$ の要素 α は，一般に 2^r 個の平方根を持つ。

$n = 2^2 p_1^{e_1} \cdots p_r^{e_r}$（$p_i$ は奇素数）と素因数分解される場合は $\#((\mathbb{Z}/n\mathbb{Z})^\times)^2 = \dfrac{\varphi(n)}{2^{r+1}}$ であり，$((\mathbb{Z}/n\mathbb{Z})^\times)^2$ の要素 α は，一般に 2^{r+1} 個の平方根を持つ。

$n = 2^{e_0} p_1^{e_1} \cdots p_r^{e_r}$ $(e_0 \geqq 3)$（p_i は奇素数）と素因数分解される場合は $\#((\mathbb{Z}/n\mathbb{Z})^\times)^2 = \dfrac{\varphi(n)}{2^{r+2}}$ であり，$((\mathbb{Z}/n\mathbb{Z})^\times)^2$ の要素 α は，一般に 2^{r+2} 個の平方根を持つ。

3.4.5 奇素数を法とする -1 の平方根

p が 4 で割ったときに 1 余るような素数であれば，-1 は p を法とする平方根を持ち，その平方根は簡単に求めることができる。

定理 3.113. p を奇素数とする。-1 が p を法とする平方剰余であるための必要十分条件は $p \equiv 1 \pmod 4$ である。

証明: オイラーの基準より -1 が p を法とする平方剰余であるための条件は $(-1)^{(p-1)/2} \equiv 1 \pmod p$ である。p は奇素数なので，$p \equiv 1 \pmod 4$ または $p \equiv 3 \pmod 4$ であるが，前者の場合 $(p-1)/2$ は偶数だから $(-1)^{(p-1)/2} = 1$ であり，後者の場合 $(p-1)/2$ は奇数だから $(-1)^{(p-1)/2} = -1$ だ。 □

定理 3.114 (p を法とする -1 の平方根)**.** p が素数であり $p \equiv 1 \pmod 4$ とする。γ を p を法とする平方非剰余とするとき，$\beta = \gamma^{(p-1)/4}$ とおけば $\beta^2 = -1$，すなわち β は -1 の平方根である。

証明: オイラーの基準と簡単な計算による。 □

補題 3.115 (Thue の補題)**.** $n, R, T \in \mathbb{Z}$ を $0 < n < RT$ なる整数とする。任意の整数 $b \in \mathbb{Z}$ について，次のような整数 $r, t \in \mathbb{Z}$ が存在する。

$$r \equiv bt \pmod n, |r| < R, 0 < |t| < T$$

証明: $i \in [0..R)$ と $j \in [0..T)$ に対して，$i - bj$ という RT 個の整数を考えると，部屋割り論法により，$i_1 - bj_1 \equiv i_2 - bj_2 \pmod n$ であるような異なる (i_1, j_1) と (i_2, j_2) が存在する。$r = i_1 - i_2$，$t = j_1 - j_2$ とおくと，r と t は上の補題の要件を満たす。 □

定理 3.116 (フェルマの 2 平方定理)**.** p を奇素数とする。$p \equiv 1 \pmod 4$ であるとき，かつそのときに限り，整数 $r, t \in \mathbb{Z}$ が存在して $p = r^2 + t^2$ と書ける。

証明: $p \equiv 1 \pmod 4$ でないとする。すると p は奇数だから，$p \equiv 3 \pmod 4$ である。奇数の平方は 4 を法として 1 と合同であり，偶数の平方は 4 を法として 0 と合同である。従って，2 つの平方数の和が 4 を法として 3 と合同な数 p になることはない。

逆に $p \equiv 1 \pmod 4$ とする。定理 3.113 により -1 は p を法とする平方剰余だから，$-1 \equiv b^2 \pmod p$ なる b が存在する。上の Thue の補題で $R = T = \lceil \sqrt{p} \rceil$，$n = p$ とすると，これらは補題の条件を満たすから，

$$r \equiv bt \pmod p, |r| < \lceil \sqrt{p} \rceil, 0 < |t| < \lceil \sqrt{p} \rceil$$

なる r と t が存在する。従って

$$r^2 \equiv b^2 t^2 \equiv -t^2 \pmod p$$

であるが，$0 < r^2 + t^2 < 2p$ だから $r^2 + t^2 = p$ でなければならない。 □

次の 2 つの定理は,「ディリクレの算術級数定理」と呼ばれるものの特殊な場合である。ディリクレの算術級数定理とは,「初項と公差が互いに素である等差数列には，素数が無限個に含まれる」というもので，一般の場合の証明はずっと厄介だが，公差が 4 の場合は，以下のように容易に示される。

定理 3.117. $p \equiv 1 \pmod 4$ なる素数 p は無限個存在する。

証明: $p \equiv 1 \pmod 4$ なる素数 p は，有限個しかないとし，それらを $p_1, \ldots, p_k$ とする。$4p_1^2 \cdots p_k^2 + 1$ を割り切る素数 p を考えると，p は奇数であり $(2p_1 \cdots p_k)^2 \equiv -1 \pmod p$ だから，-1 は p を法とする平方剰余である。よって，定理 3.113 より $p \equiv 1 \pmod 4$ でなければならない。明らかに p は $p_1, \ldots, p_k$ のどれでもないが，これはそれ以外に $p \equiv 1 \pmod 4$ なる素数が無いとしたことに矛盾する。 □

定理 3.118. $p \equiv 3 \pmod 4$ なる素数 p は無限個存在する。

証明: $p \equiv 3 \pmod 4$ なる素数 p は，有限個しかないとし，それらを $p_1, \ldots, p_k$ とする。$N = 4p_1 \cdots p_k - 1$ とすると，$N \equiv 3 \pmod 4$ だから，N を割り切る素数の中に $p \equiv 3 \pmod 4$ なるものが存在する。明らかに p は $p_1, \ldots, p_k$ のどれでもないが，これはそれ以外に $p \equiv 3 \pmod 4$ なる素数が無いとしたことに矛盾する。 □

問題 3.119. n を正の整数とする。任意の整数 m について，次が成り立つ。

(a) $m \perp \varphi(n)$ ならば $((\mathbb{Z}/n\mathbb{Z})^\times)^m = (\mathbb{Z}/n\mathbb{Z})^\times$ である。
(b) $\alpha \in ((\mathbb{Z}/n\mathbb{Z})^\times)^m$ ならば $\alpha^{\varphi(n)/\gcd(m,\varphi(n))} = 1$ である。

問題 3.120. $p = 11$ の場合に，ウィルソンの定理（定理 3.100) 証明に用いた 2 点集合 $\{\kappa, \lambda\}$ に，$(\mathbb{Z}/p\mathbb{Z})^\times$ を分割せよ。また，$\alpha = -1$ および $\alpha = 2$ の場合に，オイラーの基準（定理 3.101) の (3) の証明に用いた 2 点集合 $\{\kappa, \lambda\}$ に，$(\mathbb{Z}/p\mathbb{Z})^\times$ を分割せよ。

問題 3.121. $4, 8, 16$ を法とする 1 の平方根を全て求めよ。

問題 3.122. p を $p \equiv 1 \pmod 4$ なる素数とする。$b = ((p-1)/2)!$ とするとき，$b^2 \equiv -1 \pmod p$ である。

問題 3.123. p, q を異なる素数とする。$\alpha\beta \in (\mathbb{Z}/pq\mathbb{Z})^\times$ で $\alpha, \beta \notin (\mathbb{Z}/pq\mathbb{Z})^\times$ となる $\alpha, \beta \in \mathbb{Z}/pq\mathbb{Z}$ が存在することを示せ。

問題 3.124. p を $p \equiv 3 \pmod 4$ なる素数とすると，$((\mathbb{Z}/p\mathbb{Z})^\times)^2 = ((\mathbb{Z}/p\mathbb{Z})^\times)^4$ である。

問題 3.125. p を奇素数とし，e は 2 以上の整数とする。任意の整数 a について，$z^2 \equiv a \pmod{p^e}$ が整数解 z を持つための必要十分条件は，a を $a = p^f b$ ($b \not\equiv 0 \pmod p$) と分解したとき，次のどちらかが成り立つことである。

(a) $f \geq e$
(b) f が e より小さい偶数で，b が p を法とする平方剰余である。

(a) の場合，解 z は p^e を法として $p^{\lfloor e/2 \rfloor}$ 個存在する。(b) の場合，解 z は p^e を法として $2p^{f/2}$ 個存在する。

問題 3.126. u と v がともに 2 つの平方数の和に書けるなら，積 uv も 2 つの平方数の和に書ける。

問題 3.127. n が 2 つの平方数の和に $u^2 + v^2$ と書け，素数 p で割り切れるとする。

(a) p が u も v も割り切らないなら，$p \equiv 1 \pmod 4$ である。
(a) p は u を割り切るなら，v も割り切り，従って p^2 は n を割り切る。

問題 3.128. 正の整数 n の無平方分解を ab^2 とする。n が 2 つの平方数の和に書けるための必要十分条件は，$p \equiv 3 \pmod 4$ なる素因数 p を a が含まないことだ。

3.5 ディリクレ積とメビウス関数★

定義 3.129. 正の整数全体から実数への関数を**算術関数**（arithmetic function）と呼ぶ[4]。算術関数 f と g に対して，次で定義される算術関数 $f \star g$ を f と g の**ディリクレ積**（Dirichlet product）という。

$$(f \star g)(n) = \sum_{d \backslash n} f(d) g(n/d)$$

ここで和は n の約数 d 全部に亘る。

$$(f \star g)(n) = \sum_{d_1 d_2 = n} f(d_1) g(d_2)$$

と書くこともできる。この場合，和は $d_1 d_2 = n$ を満たす正の整数 d_1 と d_2 の対全体に亘る。

ディリクレ積は明らかに可換であり，結合法則を満たす。特に 3 つ以上の積は

$$(f_1 \star \cdots \star f_k)(n) = \sum_{d_1 \cdots d_k = n} f_1(d_1) \cdots f_k(d_k)$$

と書ける。

[4]実は，算術関数の値域としては，複素数全体 $\mathbb{C}$ を考えることもあるが，この講義ノートの範囲ではその必要がないから，簡単のため実数とする

算術関数 I，$\mathbf{1}$，μ を次のように定義する。μ はメビウス（Möbius）関数と呼ばれる。

$$I(n) = [n = 1]$$
$$\mathbf{1}(n) = 1$$
$$\mu(n) = \begin{cases} 0 & n \text{ が無平方でないとき} \\ (-1)^r & n \text{ が } r \text{ 個の異なる素数に素因数分解されるとき} \end{cases}$$

次の性質は基本的である。任意の算術関数に対して，

$$I \star f = f, \quad (\mathbf{1} \star f)(n) = \sum_{d \backslash n} f(d),$$

すなわち，I はディリクレ積の単位元であり，$\mathbf{1}$ とのディリクレ積は，全約数に亘って総和をとる演算に相当する。

定義 3.130. 算術関数 f は，$f(1) = 1$ を満たし，$n \perp m$ なる整数 m, n に対して常に $f(mn) = f(m)f(n)$ となるなら**乗法的**（multiplicative）という。

関数 I，$\mathbf{1}$，μ は乗法的である。また，オイラーの関数 φ や恒等関数 $\mathbf{id}(n) = n$ も乗法的である。

定理 3.131. 乗法的な算術関数は，素数ベキに対する値が定まれば，一意に定まる。すなわち，f を乗法的な算術関数とし，n の素因分解を $n = p_1^{e_1} \cdots p_k^{e_k}$ とすると，

$$f(n) = f(p_1^{e_1}) \cdots f(p_k^{e_k})$$

である。ν を用いて，言い換えれば $f(n) = \prod_p f(p^{\nu_p(n)})$ である。

定理 3.132. f を乗法的な算術関数とする。n の素因分解を $n = p_1^{e_1} \cdots p_k^{e_k}$ とすると，

$$\sum_{d \backslash n} \mu(d) f(d) = (1 - f(p_1)) \cdots (1 - f(p_k))$$

である。

証明: $d \setminus n$ なる d は，$d = p_1^{d_1} \cdots p_k^{d_k}$ $(0 \leqq d_i \leqq e_i)$ の形に書けるが，ある i で $d_i > 1$ ならば，$\mu(d) = 0$ である。したがって左辺の和は，f が乗法的なことより

$$\sum_{d_1, \ldots, d_k \in \{0,1\}} \mu(p_1^{d_1} \cdots p_k^{d_k}) f(p_1)^{d_1} \cdots f(p_k)^{d_k}$$

となるが，これは右辺を展開したものに他ならない。 □

定理 3.133. $\mathbf{1} \star \mu = I$ である。すなわち，$\mathbf{1}$ と μ はディリクレ積に関して互いの逆元である。

証明: 定理 3.132 で $f = \mathbf{1}$ とし，$n > 1$ の素因分解を $n = p_1^{e_1} \cdots p_k^{e_k}$ $(k > 0)$ とすると，

$$(\mathbf{1} \star \mu)(n) = \sum_{d \backslash n} \mu(d) = \underbrace{(1-1)(1-1)\ldots(1-1)}_{k} = 0$$

である。もちろん $(\mathbf{1} \star \mu)(0) = 1$ だから，$(\mathbf{1} \star \mu)(n) = [n = 1] = I(n)$ となる。 □

定理 3.134 (メビウス反転公式)**.** 任意の算術関数 f と F について

$$F = \mathbf{1} \star f \iff f = \mu \star F$$

である。すなわち，次が成り立つ。

$$\forall n \in \mathbb{Z}\ (n > 0 \implies F(n) = \sum_{d \backslash n} f(d))$$
$$\iff \forall n \in \mathbb{Z}\ (n > 0 \implies f(n) = \sum_{d \backslash n} \mu(d) F(n/d))$$

定理 3.135. 正の整数 n に対して $\sum_{d\backslash n} \varphi(d) = n$ である。すなわち $\mathbf{1} \star \varphi = \mathbf{id}$ である。
証明: メビウス反転公式より，$\varphi = \mu \star \mathbf{id}$ を示せばいい。n の素因分解を $n = p_1^{e_1} \cdots p_k^{e_k}$ とすると，である。定理 3.132 で $f(n) = 1/n$ とすれば，

$$(\mu \star \mathbf{id})(n) = \sum_{d \backslash n} \mu(d)\mathbf{id}(n/d) = n \sum_{d \backslash n} \mu(d)/d = n(1 - 1/p_1) \cdots (1 - 1/p_k) = \varphi(n)$$

である。 □

問題 3.136. $f \in \mathbb{Z}[x]$ を整数係数の多項式とし，任意の正の整数 n に対して，$\mathbb{Z}/n\mathbb{Z}$ における方程式 $f(x) = 0$ の解の個数を $\omega_f(n)$ をするとき，ω_f は乗法的である。

問題 3.137. f と g が乗法的な算術関数とすると，$f \star g$ も乗法的である。

問題 3.138. 整数 n の正の約数の個数を $\tau(n)$ とする。

(a) τ は乗法的である。
(b) 正の整数 n の素因数分解を $n = p_1^{e_1} \cdots p_r^{e_r}$ とするとき $\tau(n) = (e_1 + 1) \cdots (e_r + 1)$ である。言い換えれば $\tau(n) = \prod_p (\nu_p(n) + 1)$ である。
(c) $\sum_{d\backslash n} \mu(d)\tau(n/d) = 1$
(d) 正の整数 n の素因数分解を $n = p_1^{e_1} \cdots p_r^{e_r}$ とするとき $\sum_{d\backslash n} \mu(d)\tau(d) = (-1)^r$ である。言い換えれば $\sum_{d\backslash n} \mu(d)\tau(d) = \prod_p (-1)^{[\nu_p(n)>0]}$ である。

問題 3.139. $\sigma(n)$ を n の正の約数の総和，すなわち $\sigma(n) \stackrel{\text{def}}{=} \sum_{d\backslash n} d$ とする。

(a) σ は乗法的である。
(b) n の素因数分解を $n = \pm p_1^{e_1} \cdots p_r^{e_r}$ とするとき $\sigma(n) = (p_1^{e_1+1} - 1)/(p_1 - 1) \cdots (p_r^{e_r+1} - 1)/(p_r - 1)$ である。言い換えれば $\sigma(n) = \prod_p (p^{\nu_p(n)+1} - 1)/(p - 1)$ である。
(c) $\sum_{d\backslash n} \mu(d)\sigma(n/d) = n$
(d) 正の整数 n の素因数分解を $n = p_1^{e_1} \cdots p_r^{e_r}$ とするとき $\sum_{d\backslash n} \mu(d)\sigma(d) = (-1)^r p_1 \cdots p_r$ である。言い換えれば $\sum_{d\backslash n} \mu(d)\tau(d) = \prod_p (-p)^{[\nu_p(n)>0]}$ である。

問題 3.140. Mangoldt 関数 Λ は任意の正の整数 n に対して，n が素数 p のべき乗のとき $\Lambda(n) = \log p$，そうでないとき $\Lambda(n) = 0$ として定義される。言い換えれば，$\Lambda(n) \stackrel{\text{def}}{=} \sum_p \sum_{k=1}^{\infty} [n = p^k] \log p$ である。$\sum_{d\backslash n} \Lambda(d) = \log n$ である。また，$\Lambda(n) = -\sum_{d\backslash n} \mu(d) \log d$ である。

問題 3.141. f を乗法的な算術関数とする。正の整数 n の素因数分解を $n = p_1^{e_1} \cdots p_r^{e_r}$ とすると $\sum_{d \backslash n} \mu(d)^2 f(d) = (1+f(p_1)) \cdots (1+f(p_r))$ である。言い換えれば $\sum_{d \backslash n} \mu(d)^2 f(d) = \prod_p (1+f(p))^{[\nu_p(n)>0]}$ である。

問題 3.142. 正の整数 n は無平方のとき，かつそのときに限り，$\sum_{d \backslash n} \mu(d)^2 \varphi(d) = n$ を満たす。

問題 3.143. どんな算術関数 f にも $f(1) \neq 0$ ならば $f \star g = I$ となる算術関数 g がただ 1 つ存在する。それを f の**ディリクレ逆元**（Dirichlet inverse）と呼ぶ。$f(1) = 0$ ならば，f にはディリクレ逆元は存在しない。

問題 3.144. f が乗法的なら，f のディリクレ逆元も乗法的である。

4 ユークリッド互除法（The Euclidean Algorithm）

本節ではユークリッドの互除法とその応用を扱う。このアルゴリズムは古代ギリシアの時代に既に知られており，ユークリッドが書いたとされる数学書「原論」に述べられている。もともとは 2 つの自然数の間の最大公約数を求めるアルゴリズムであるが，ここでは，より現代的に一般のユークリッド整域において使えるような形で提示する。ユークリッド整域としてはさまざまなものが考えられるが，代表的なのは整数環 $\mathbb{Z}$ と多項式環 $K[x]$ である。

最初に準備として，一般の可換環のイデアルや剰余環について，簡単に復習する。

4.1 イデアル，剰余環，GCD，LCM ☆

定義 4.1. R を可換環とする。その空でない部分集合 I は，次を満たすとき，R の**イデアル**（ideal）と呼ぶ。

(1) $\forall a, b \in I\ a - b \in I$
(2) $\forall a \in I\ \forall z \in R\ az \in I$

一般に，R の部分集合 M に対して，

$$\langle M \rangle \stackrel{\text{def}}{=} \{a_1 r_1 + \cdots + a_k r_k \mid a_1, \ldots, a_k \in M, r_1, \cdots, r_k \in R\}$$

は，R のイデアルになり，M が**生成する**（generate）するイデアルと呼ぶ。特に，M が有限集合 $M = \{a_1, \ldots, a_r\}$ のとき，$\langle M \rangle$ は，

$$\{a_1 z_1 + \ldots + a_r z_r \mid z_1, \ldots, z_r \in R\}$$

となり，$a_1 R + \cdots + a_r R$ と表記される。このように有限集合が生成するイデアルを**有限生成**（finitely generated）であるという。特に $r = 1$ のとき，すなわち生成元が 1 つのイデアルを**単項**（principal）イデアルと呼ぶ。また，$f \in R$ とイデアル $I = \langle M \rangle$ に対し，集合

$$f + I \stackrel{\text{def}}{=} \{f + m \mid m \in I\}$$

を $I = \langle M \rangle$ による f の**剰余類**（residue class）と呼び，$f \bmod M$ とも記す。

$$R/I \stackrel{\text{def}}{=} \{f + I \mid f \in R\}$$

は，演算を自然に定義することでそれ自身環になり，$I = \langle M \rangle$ による**剰余環**（residue class ring）と呼ぶ。また，$f - g \in I = \langle M \rangle$ であることを，$f \equiv g \pmod{I}$，$f \equiv g \pmod{M}$ などと記す。

一要素から生成されるイデアル，すなわち $a \in R$ が存在して aR の形に書けるイデアルを**単項**（principal）イデアルと呼ぶ。I が単項イデアル aR のとき，f の剰余類 $f + I$ は $\{f + ar \mid r \in R\}$ であり，$f \bmod a$ と記す。同様に $f \equiv g \pmod{a}$ という記法も用いる。

一般に，S と T を環 R の部分集合，$a \in R$ とするとき，$S + T \stackrel{\text{def}}{=} \{s + t \mid s \in S, t \in T\}$，$ST = \{s_1 t_1 + s_2 t_2 + \cdots + s_k t_k \mid s_1, s_2, \ldots, s_k \in S, t_1, t_2, \ldots, t_k \in T\}$，$aS \stackrel{\text{def}}{=} \{as \mid s \in S\}$ と定義する。

問題 4.2. I_1, I_2 が R のイデアルで，$a \in R$ のとき，$I_1 + I_2$, $I_1 \cap I_2$, $I_1 I_2$，aI_1 はイデアルである。

定義 4.3 (約元，倍元，GCD，LCM，単元，既約元)**.**
R を可換環とする。$a \in R$ が $b \in R$ を割り切る，すなわち $b = ca$ なる要素 $c \in R$ が存在するとき，$a \setminus b$ と記す。このとき a を b の**約元**（divisor）といい，反対に b を a の**倍元**（multiple）という。$a_1, \ldots, a_n \in R$ とするとき，次が成り立てば，b を $a_1, \ldots, a_n$ の **GCD**（最大公約元，greatest common divisor）という。

- $b \setminus a_1 \wedge \cdots \wedge b \setminus a_n$
- $\forall d \in R\ (d \setminus a_1 \wedge \cdots \wedge d \setminus a_n \to d \setminus b)$

対称的に，次が成り立つとき，b を $a_1, \ldots, a_n$ の **LCM**（最小公倍元，least common multiple）という。

- $a_1 \setminus c \wedge \cdots \wedge a_n \setminus c$
- $\forall d \in R\ (a_1 \setminus d \wedge \cdots \wedge a_n \setminus d \to c \setminus d)$

$u \in R$ は，1 の約元であるとき，すなわち $uv = 1$ を満たす $v \in R$ が存在するとき，**単元**（unit）と呼ぶ。また，$1 = uv$ を満たす v を u^{-1} と記し，u の（乗法）**逆元**（inverse）と呼ぶ。$u = u_1 u_2$ $(u_1, u_2 \in R)$ と分解するといつでも u_1 か u_2 のどちらかが単元になるとき，u を**既約**（irreducible）という。ある単元 u が存在して，$a = ub$ となるとき $a \sim b$ と記す。

GCD は，一般には一意ではない。しかし，c と d がともに，$a_1, \ldots, a_n$ の GCD ならば $c \sim d$ である。特に $R = \mathbb{Z}$ の場合，GCD，LCM は符号を除けば同じなので，その絶対値と定義し，$a_1, \ldots, a_n \in \mathbb{Z}$ の GCD, LCM をそれぞれ $\gcd(a_1, \ldots, a_n)$，$\mathrm{lcm}(a_1, \ldots, a_n)$ と記すことにしたわけである。

問題 4.4. $\sim$ は R 上の同値関係である。

問題 4.5. 次の同値性を証明せよ。

$$a \sim b \iff (a \setminus b \wedge b \setminus a) \iff aR = bR$$

ここで，aR は a が生成する単項イデアル，すなわち $\{ax \mid x \in R\}$ を表す。

問題 4.6. $\mathbb{Z}$，$K[x]$，$\mathbb{Z}[\mathrm{i}]$ の単元は，それぞれどういうものか。ただし，K は任意の体とし，i は虚数単位（-1 の平方根），すなわち $\mathrm{i} = \sqrt{-1}$ とする。従って $\mathbb{Z}[\mathrm{i}]$ は，$m + n\mathrm{i}$ $(m, n \in \mathbb{Z})$ という形の複素数の全体であり，そのような複素数は**ガウス整数**（Gaussian integer）と呼ばれる。

問題 4.7. I は可換環 R のイデアルで，$a, b \in R$ とする。次を示せ。

(1) $b \in aR \iff a \setminus b$
(2) $b \in I \iff bR \subset I$
(3) $bR \subset aR \iff a \setminus b$

問題 4.8. 整数環 $\mathbb{Z}$ の部分集合 $I \subset \mathbb{Z}$ が空でなく，減法について閉じている（すなわち任意の $a, b \in I$ について $a - b \in I$）とする。このとき I は $\mathbb{Z}$ のイデアルであることを示せ。

問題 4.9. $\mathbb{Z}$ のイデアル $I = a\mathbb{Z}$，$J = b\mathbb{Z}$ について $I \cap J = \mathrm{lcm}(a, b)\mathbb{Z}$ が成り立つ。

問題 4.10. $I_1 = 20\mathbb{Z}$，$I_2 = 30\mathbb{Z}$，$I_3 = 50\mathbb{Z}$ とする。次のイデアルを単項イデアルとして表現せよ。

(a) $I_1 + I_2 + I_3$
(b) $I_1 \cap I_2 \cap I_3$
(c) $I_1 + I_2 I_3$

4.2 ユークリッド整域と互除法

体 K の要素を係数に持つ 1 変数多項式の全体 $K[x]$ と整数の全体 $\mathbb{Z}$ とは，多くの類似点を持つ。具体的にいうと，ともに整域であり，剰余つき割り算を定義することができる。これは，これらが共にユークリッド整域と呼ばれる代数構造をなしているということからくる帰結である。

まず，前の章で述べた整数上の諸概念をユークリッド整域に拡張することから始めよう。まず，定理 3.8 と定義 3.9 を拡張する。

定義 4.11 (ユークリッド整域)**.**
R を整域，d を R から $\mathbb{N}$ への関数とする。このとき，任意の $a, b \in R$ に対して，$b \neq 0$ ならば

$$a = qb + r, \quad d(r) < d(b)$$

を満たす $q, r \in R$ が存在するとき，(R, d) を**ユークリッド整域**（Euclidean domains）と呼ぶ。このような関数 d を**ユークリッド関数**（Euclidean function）と呼ぶ。また，上の q を（a を b で割ったときの）**整商**（quotient，あるいは単に商）と呼び，r を**剰余**（remainder） と呼ぶが，与えられた a と b に対し，整商や剰余がただ 1 つに定まるとは限らない。

上の用語から分かるようにユークリッド整域とは，剰余つき除算が可能な整域と考えてよい。ユークリッド整域では，$b \neq 0$ ならば，$d(r) < d(b)$ なる r が存在しなければならないので，$d(b) \neq 0$ である。

例 4.12. ユークリッド整域の例。

(1) $\mathbb{Z}$ は，ユークリッド関数 d を $d(n) \stackrel{\text{def}}{=} |n|$ で定義すれば，ユークリッド整域である。
(2) K を任意の体とするとき，$K[x]$ は，ユークリッド関数 d を $d(0) \stackrel{\text{def}}{=} 0, d(f) \stackrel{\text{def}}{=} \deg f + 1$ $(f \neq 0)$ で定義すれば，ユークリッド整域である。
(3) $\mathbb{Z}[\mathrm{i}]$ は，ユークリッド関数 d を $d(m + n\mathrm{i}) \stackrel{\text{def}}{=} m^2 + n^2$ で定義すれば，ユークリッド整域である。

問題 4.13. 上の例の (3)，すなわち $(\mathbb{Z}[\mathrm{i}], d)$ がユークリッド整域になることを証明せよ。

定義 4.14 (互いに素)**.** $a_1, \ldots, a_n$ の GCD が単元のとき，$a_1, \ldots, a_n$ は**互いに素**（mutually prime, relatively prime, coprime）という。特に a, b が互いに素であるとき，$a \perp b$ と記す。

GCD と LCM は，一般の環 R においては，必ず存在すると限らないが，R がユークリッド整域であれば必ず存在し，それを計算するアルゴリズムも存在する。ここで R はユークリッド整域で，剰余つき乗算のアルゴリズムは与えられているものとし，$(q, r) \longleftarrow a \div b$ でそれを表す。すなわち，任意の a, b に対し，$b \neq 0$ ならば $a = qb + r$ かつ $d(r) < d(b)$ なる q, r が計算されるものとする。

ユークリッド互除法（Euclidean Algorithm）

入力 $f, g \in R$
出力 f と g の GCD $d \in R$

$EA(f, g)$
(1) $r_0 \longleftarrow f$; $r_1 \longleftarrow g$; $i \longleftarrow 1$;
(2) **while** $r_i \neq 0$ **do**
$(q_i, r_{i+1}) \longleftarrow r_{i-1} \div r_i$; $i \longleftarrow i + 1$;
(3) $d \longleftarrow r_{i-1}$;

定理 4.15. 上のアルゴリズムは必ず停止し，その出力 d は a と b の GCD である。
証明: アルゴリズムが停止しなければ，$d(r_1) > d(r_2) > \cdots$ と無限に減少していく列が $\mathbb{N} \cup \{-\infty\}$ の中に得られる。これは矛盾である。

また $r_{i-1} = q_i r_i + r_{i+1}$ より，r_{i-1} と r_i の GCD は r_i と r_{i+1} との GCD に等しい。よって，r が a と b の GCD であることは，$r_0 = a$，$r_1 = b$ と，アルゴリズム停止したときは $r_i = 0$ であるから r_{i-1} と r_i との GCD は $d = r_{i-1}$ となっていることからわかる。 □

例 4.16. 例えば $R = \mathbb{Z}$，$f = 119$，$g = 35$ の場合に，アルゴリズムを適用してみると次のように計算が進み，結果 d として $r_3 = 7$ が返される。

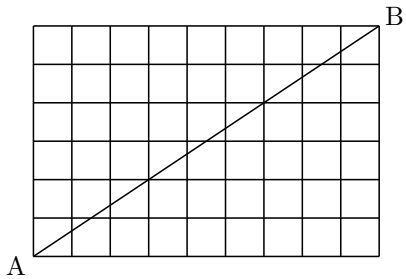
i	0	1	2	3	4
r_i	119	35	14	7	0
q_i		3	2	2	

例 4.17. 同様に $R = \mathbb{Q}[x]$，$f = 18x^3 - 42x^2 + 30x - 6$，$g = -12x^2 + 10x - 2$ の場合は，

i	0	1	2	3
r_i	$18x^3 - 42x^2 + 30x - 6$	$-12x^2 + 10x - 2$	$(9/2)x - 3/2$	0
q_i	$-(3/2)x + 9/4$	$-(8/3)x + 4/3$		

により，結果 d は $r_2 = \dfrac{9}{2}x - \dfrac{3}{2}$ となる。

問題 4.18. 長方形の格子模様がある。下図のような 6×9 の格子の場合，対角線 AB が格子線を横切る回数は（数えれば分かるように）11 回である。7785×8477 の格子の場合はどうか？



問題 4.19. 7785×8477 の長方形をしたビリヤード台がある。この左下隅から右上向きに正確に $45°$ の角度でボールを衝いたとすると，ボールは四隅のどこに到達し，それまでに何回辺で反射するか？各辺でのボールの反射は，入射角と反射角が正確に同じであり，いっさい減速しないものとする。

問題 4.20. R をユークリッド整域，d をそのユークリッド関数とする。$r \neq 0$ ならば，$d(r) > d(0)$ であることを示せ。また，$d(ab) = d(a)$ となるための必要十分条件は b が単元のときであることを示せ。

問題 4.21. ユークリッド整域において LCM を計算するアルゴリズムを与えよ。

問題 4.22. $R = \mathbb{Z}$ の場合に，整数が 2 進表記されていると，次の**2 進 gcd アルゴリズム**により，乗除算といった複雑な演算を避けシフト演算を用いることで，最大公約数をさらに高速に計算することができる。

2 進 gcd アルゴリズム（binary gcd Algorithm）

入力 2 進表記された正の整数 f, g
出力 f と g の 2 進表記された最大公約数 d

(1) $r \longleftarrow f$; $s \longleftarrow g$; $e \longleftarrow 0$
(2) **while** $2 \setminus r \wedge 2 \setminus s$ **do**
$r \longleftarrow r \gg 1$; $s \longleftarrow s \gg 1$; $e \longleftarrow e + 1$;
(3) **repeat**
while $2 \setminus r$ **do** $r \longleftarrow r \gg 1$; **while** $2 \setminus s$ **do** $s \longleftarrow s \gg 1$;
if $s < r$ **then** $r \longleftarrow r - s$ **else** $s \longleftarrow s - r$;
until $s = 0$
(4) $d \longleftarrow r \ll e$;

ここで $a \ll e$ と $a \gg e$ は，それぞれ a を e ビット左または右へシフトすることを表す。すなわち，実質的には $a \cdot 2^e$ と $a \div 2^e$ と同じである。

上のアルゴリズムで正しく最大公約数が計算できることを証明せよ。

4.3 拡張ユークリッド互除法

ユークリッド整域 R において GCD が重要なのは，$a_1, \ldots, a_n$ の GCD がそれらの線形結合によって表せるからである。すなわち，c が $a_1, \ldots, a_n$ の GCD であれば，$b_1, \ldots, b_n \in R$ が存在して，$a_1 b_1 + \cdots + a_n b_n = c$ である。この $b_1, \ldots, b_n$ を求めるアルゴリズムも存在する。

拡張ユークリッド互除法（Extended Euclidean Algorithm）

入力 $f, g \in R$
出力 f と g の GCD $d \in R$，$fs + gt = d$ となる $s, t \in R$

EEA(f, g)
(1) $r_0 \longleftarrow f$; $s_0 \longleftarrow 1$; $t_0 \longleftarrow 0$; $r_1 \longleftarrow g$; $s_1 \longleftarrow 0$; $t_1 \longleftarrow 1$; $i \longleftarrow 1$;
(2) **while** $r_i \neq 0$ **do**
$(q_i, r_{i+1}) \longleftarrow r_{i-1} \div r_i$; $s_{i+1} \longleftarrow s_{i-1} - q_i s_i$; $t_{i+1} \longleftarrow t_{i-1} - q_i t_i$; $i \longleftarrow i + 1$;
(3) $d \longleftarrow r_{i-1}$; $s \longleftarrow s_{i-1}$; $t \longleftarrow t_{i-1}$;

このアルゴリズムは出力として組 $\langle d, s, t\rangle$ を返すので，それらを受け取る場合には $\langle d, s, t\rangle \longleftarrow \mathrm{EEA}(f, g)$ のように表記することにする。しかし，あとの応用で見るように，$\langle d, s, t\rangle = \langle r_{i-1}, s_{i-1}, t_{i-1}\rangle$ だけでなく，上記の計算で得られる列 $\langle r_0, s_0, t_0\rangle, \langle r_1, s_1, t_1\rangle, \ldots, \langle r_{i-1}, s_{i-1}, t_{i-1}\rangle$ 全体が有用なことも多い。

例 4.23. 例えば $R = \mathbb{Z}$，$f = 119$，$g = 35$ の場合に，アルゴリズムを適用してみると次のように計算が進み，d, s, t として，$r_3 = 7, s_3 = -2, t_3 = 7$ が返されるが，実際 $7 = 119 \cdot (-2) + 35 \cdot 7$ である。

i	q_i	r_i	s_i	t_i
0		119	1	0
1	3	35	0	1
2	2	14	1	-3
3	2	7	-2	7
4	2	0	5	-17

例 4.24. $R = \mathbb{Q}[x]$，$f = 18x^3 - 42x^2 + 30x - 6$，$g = -12x^2 + 10x - 2$ の場合は，

i	q_i	r_i	s_i	t_i
0		$18x^3 - 42x^2 + 30x - 6$	1	0
1	$-(3/2)x + 9/4$	$-12x^2 + 10x - 2$	0	1
2	$-(8/3)x + 4/3$	$(9/2)x - 3/2$	1	$(3/2)x - 9/4$
3		0	$(8/3)x - 4/3$	$4x^2 - 8x + 4$

と計算が進み，r, s, t として，$r_2 = \dfrac{9}{2}x - \dfrac{3}{2}$，$s_2 = 1$，$t_2 = \dfrac{3}{2}x - \dfrac{9}{4}$ が返されるが，実際

$$\frac{9}{2}x + \frac{3}{2} = (18x^3 - 42x^2 + 30x - 6) \cdot 1 + (-12x^2 + 10x - 2)\left(\frac{3}{2}x - \frac{9}{4}\right)$$

である。

このアルゴリズムが正しいことは，$r_i = s_i f + t_i g$ が常に成り立つことから分かるが，より精密にアルゴリズムの特徴を調べると，次のようなことが分かる。

定理 4.25. アルゴリズム内の変数 r_i, t_i, s_i を用いて

$$T_i \stackrel{\mathrm{def}}{=} \begin{pmatrix} s_i & s_{i+1} \\ t_i & t_{i+1} \end{pmatrix}, \quad Q_i \stackrel{\mathrm{def}}{=} \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix}$$

と定義する。このとき，次が成り立つ。

(a) $(f\ g)\begin{pmatrix} s_i \\ t_i \end{pmatrix} = r_i$，従って $(f\ g)T_i = (r_i\ r_{i+1})$ である。
(b) $T_i = T_0 Q_1 \cdots Q_i = Q_1 \cdots Q_i$
(c) r は，(任意の i について) r_i と r_{i+1} の GCD である。従って，特に $f = r_0$ と $g = r_1$ の GCD である。
(d) $\det T_i = s_i t_{i+1} - t_i s_{i+1} = (-1)^i$，従って $s_i \perp t_i$ である。
(e) $f = (-1)^i(t_{i+1} r_i - t_i r_{i+1})$，$g = (-1)^i(-s_{i+1} r_i + s_i r_{i+1})$

証明: (a), (b) は i に関する帰納法による。(c) は通常の互除法の場合と同様である。(d) は，$\det Q_i = -1$，$\det T_0 = 1$ と (b) より得られる。(e) は，(a) と (d) より（クラメルの公式などで方程式を解いて）得られる。 □

ユークリッド整域は，整数の全体 $\mathbb{Z}$ の持つ性質を抽象化したものであり，そのもっとも重要な帰結が上の定理である。この定理から，3 章で述べたような $\mathbb{Z}$ の持つたくさんの性質をユークリッド整域 R も持つことが分かるし，多くの計算問題を解く実際的なアルゴリズムも得られる。

以下では，$\langle R, d\rangle$ をユークリッド整域し，そこにおけるイデアルと整除性の関係について調べる。従って，$R = \mathbb{Z}$ の場合に適用することで，3 節で述べた整数の多くの性質が得られる。

定理 4.26. R においては，そのすべてのイデアル I が単項イデアルである。特に，$f, g \in R$ が生成するイデアル $fR + gR$ は，f, g の GCD d が生成する単項イデアル dR に等しい。特に，$R = \mathbb{Z}$ の場合，定理 3.17，系 3.18 が得られる。

証明: I をイデアルとする。$I = \{0\}$ なら，I は 0 が生成する単項イデアルである。$f \neq 0$ を $d(f)$ が最小の I の元とする。$I = fR$ を示そう。明らかに $fR \subset I$ である。$g \in I$ とすると，ユークリッド整域の定義により $g = qf + r$ かつ $d(r) < d(f)$ なる q, r が存在するが，$r = g - qf \in I$ だから，$d(f)$ の最小性より $r = 0$ でなければならない。ゆえに $g = qf$ だから $I \subset fR$ である。

また，$f, g \in R$ の GCD を d とすると，アルゴリズム EEA により，$fs + gt = d$ となる。$s, t \in R$ が得られる。$du = fsu + gtu$ だから $dR \subset fR + gR$，逆に d は f, g の公約元だから，$f = df', g = dg'$ とおくと，$fu + gv = d(f'u + g'v)$ より，$fR + gR \subset dR$ である。 □

系 4.27. ユークリッド整域 R においては，任意の $a, b \in R$ が常に GCD を持ち，それは $s, t \in R$ を用いて a と b の線形結合 $as + bt$ の形に書き表せる。特に $a \perp b$ ならば $1 = as + bt$ と書ける。

証明: a と b の GCD が $as+bt$ の形に書けることは，EEA からの直接の帰結である。$a \perp b$ ならば，定義より GCD は単元だから，それを u とすると $u = as+bt$ と書ける。従って $1 = uu^{-1} = a(su^{-1})+b(tu^{-1})$ である。 □

問題 4.28. ユークリッド整域 R においては，次が成り立つことを証明せよ。

(a) $a \perp b \wedge a \perp c \Longrightarrow a \perp bc$
(b) $a \perp b \wedge a \setminus bc \Longrightarrow a \setminus c$
(c) $a \perp b \wedge a \setminus c \wedge b \setminus c \Longrightarrow ab \setminus c$

問題 4.29. $\mathbb{Z}[\sqrt{p}]$ を $\{n + m\sqrt{p} \mid m, n \in \mathbb{Z}\}$ と定義する。

(a) 任意の整数 p について $\mathbb{Z}[\sqrt{p}]$ は $\mathbb{C}$ の部分環になることを示せ。
(b) $R = \mathbb{Z}[\mathrm{i}]$ において，± 1 と $\pm\mathrm{i}$ 以外には単元が存在しないことを示せ。（ヒント: $n + m\mathrm{i}$ の絶対値の 2 乗は $n^2 + m^2$ であることを利用せよ）
(c) $R = \mathbb{Z}[\mathrm{i}]$ において，3 と $1 + \mathrm{i}$ が既約元であることを示せ。
(d) $R = \mathbb{Z}[\mathrm{i}]$ において，2 は既約でないこと示せ。
(e) $R = \mathbb{Z}[\mathrm{i}]$ において，$1 + 3\mathrm{i}$ と $5 + \mathrm{i}$ の GCD の 1 つを求め，その GCD を $(1 + 3\mathrm{i})a + (5 + \mathrm{i})b$ の形に表現する $a, b \in \mathbb{Z}[\mathrm{i}]$ を求めよ。
(f) $R = \mathbb{Z}[\sqrt{2}]$ には，± 1 以外にも単元が存在することを示せ。

(g) $R = \mathbb{Z}[\sqrt{2}]$ は，ユークリッド整域であることを示せ。（ヒント: ユークリッド関数として $d(n + m\sqrt{2}) = |n^2 - 2m^2|$ を考えよ）
(h) $R = \mathbb{Z}[\sqrt{-5}]$ において，± 1 以外には単元が存在しないことを示せ。（ヒント: $n + m\sqrt{-5}$ の絶対値の 2 乗は $n^2 + 5m^2$ であることを利用せよ）
(i) $R = \mathbb{Z}[\sqrt{-5}]$ において，2 と $1 + \sqrt{-5}$ が既約元であることを示せ。
(j) $R = \mathbb{Z}[\sqrt{-5}]$ がユークリッド整域でないことを示せ。

問題 4.30. $\mathbb{Z}[x]$ がユークリッド整域ではありえないことを示せ。

問題 4.31.
$\omega = \dfrac{1 + \sqrt{-3}}{2}$ とし，$\mathbb{Z}[\omega] = \{n + m\omega \mid m, n \in \mathbb{Z}\}$ を考える。

(a) $\mathbb{Z}[\omega]$ が環であることを示せ。

(b) $\mathbb{Z}[\omega]$ がユークリッド整域であることを示せ。

(c) $\mathbb{Z}[\omega]$ の単元をすべて求めよ。

（ヒント：$\omega^2 = \omega - 1$ である。また，$\mathbb{Z}[\omega]$ は複素平面上で辺長 1 の三角格子を形成する。）

問題 4.32. $R = \mathbb{Z}$ の場合の拡張ユークリッド互除法について，$f \geqq g \geqq 0$，$r_{\lambda+1} = 0$ とするとき，次が成り立つ（記号はアルゴリズムの中と同じものを用いる）。

(a) $i = 0, 1, \ldots, \lambda$ について $t_i t_{i+1} \leqq 0$ かつ $|t_i| \leqq |t_{i+1}|$ が成り立つ。$i = 1, 2, \ldots, \lambda$ について $s_i s_{i+1} \leqq 0$ かつ $|s_i| \leqq |s_{i+1}|$ が成り立つ。
(b) $i = 1, 2, \ldots, \lambda + 1$ について $r_{i-1}|t_i| \leqq f$ かつ $r_{i-1}|s_i| \leqq g$ が成り立つ。
(c) $f > 0$ なら，$i = 1, 2, \ldots, \lambda + 1$ について $|t_i| \leqq f$ かつ $|s_i| \leqq g$ が成り立つ。$f > 1$ で $g > 0$ なら，$|t_\lambda| \leqq f/2$ かつ $|s_\lambda| \leqq g/2$ が成り立つ。
(d) $i = 0, 1, \ldots, \lambda + 1$ について，$s_i t_i \leqq 0$ が成り立つ。
(e) $d = \gcd(f, g) > 0$ なら，$|s_{\lambda+1}| = g/d$，$|t_{\lambda+1}5| = f/d$ である。
(f) $r_i = 0$ のとき $|f s_i| = |g t_i|$ であり，これは f と g の最小公倍元となる。

4.4 拡張ユークリッド互除法の計算量

拡張ユークリッド互除法はきわめて効率の良いアルゴリズムである。しかし，計算量に関する厳密な議論は複雑になるので，ここでは面倒な議論を避け，結果だけを述べる。

定理 4.33.
$f, g \in K[x]$ とし，$\deg f = n$，$\deg g = m$ とする。f, g を入力とする拡張ユークリッド互除法の計算量は，体 K 上での 1 回の加減乗除が定数時間で計算できると仮定するなら，$O(nm)$ である。すなわち，入力多項式の次数の積に比例する時間で計算できる。

定理 4.34.
$f, g \in \mathbb{Z}$ とし，$\mathrm{len}(f) = n$，$\mathrm{len}(g) = m$ とする。f, g を入力とする拡張ユークリッド互除法の計算量は，単精度整数 1 回の加減乗除が定数時間で計算できると仮定するなら，$O(nm)$ である。すなわち，入力整数の桁数の積に比例する時間で計算できる。

問題 4.35. 問題 4.21 で与えた LCM を計算するアルゴリズムの計算量を見積もれ。

問題 4.36. 問題 4.22 で与えた 2 進 gcd アルゴリズムの計算量を見積もれ。また，拡張互助法と同様，$fs+gt=\gcd(f,g)$ を満たす s,t を同時に計算するように 2 進 gcd アルゴリズムを修正せよ。

4.5 ユークリッド関数の性質について★

我々がユークリッド整域 R のユークリッド関数 $d:R\to\mathbb{N}\cup\{-\infty\}$ の性質として要求したものは

$$\forall f,g\in R\ (g\neq 0\to\exists q,r\in R\ f=qg+r\wedge d(r)\leqq d(g)) \tag{1}$$

というものだけであった。しかし，別の書物の定義などでは，さらに

$$\forall f,g\in R\ d(f)\leqq d(fg) \tag{2}$$

という条件が加わっていう場合がある。実際，例 4.12 で $\mathbb{Z}$，$K[x]$，$\mathbb{Z}$[i] に対して定義したユークリッド関数 d は，どれも (2) の性質を持つ。さらに，書物によっては，d の値域が $\mathbb{N}$ になっていることもあるし，さらに強い条件

$$\forall f,g\in R\ d(f)d(g)=d(fg) \tag{3}$$

を課すことがあるかもしれない。実際，上の $\mathbb{Z}$，$\mathbb{Z}$[i] に対する d は (3) を満足するし，$K[x]$ の場合も，2 以上の任意の整数 n 選び，$d(0)\stackrel{\text{def}}{=}0$，$d(f)\stackrel{\text{def}}{=}n^{\deg(f)}$ と d を定義し直すことで，性質 (3) を持つようになる。

しかし，これらの条件がユークリッド整域を特徴付けるものとして本質的かというと，そういうことはない。実際，定理 4.15，4.26 などの証明を見れば分かるように，ユークリッド関数 d の値域は，整列集合であれば何でもかまわない。また，どんなユークリッド整域も，自動的に (2) を満足するユークリッド関数を持つことが次のように示される。

定理 4.37. R を任意の可換環とし，$\langle X,<\rangle$ を任意の整列集合とする。R から X の関数で (1) を満足するものをユークリッド関数と呼ぶ。d をユークリッド関数とするとき，d が (2) を満たさなければ，すなわち $d(ab)<d(a)$ なる $a,b\in R$ が存在すれば，(2) を満たすユークリッド関数 d^* で $d^*<d$ なるものが存在する。

証明: R から X への関数 d^* を

$$d^*(f)=\min\{d(fh)\mid h\in R,h\neq 0\}$$

と定義する。このとき，任意の $x\in R$ について $d^*(f)\leqq d(f1)=d(f)$ であり，かつ $d^*(a)\leqq d(ab)<d(a)$ だから $d^*<d$ である。また，任意の $f,g\in R$ について

$$d^*(f)=\min\{fh\mid h\in R,h\neq 0\}\leqq\min\{fgk\mid k\in R,k\neq 0\}=d^*(fg)$$

だから d^* は (2) を満たす。最後に d^* がユークリッド関数であることを示すために，f，$g\neq 0$ を任意に取り，$d^*(g)=d(gh)$ を実現する $h\neq 0$ を選ぶと，$gh\neq 0$ であり d は (1) を満たすので，q,r が存在して $f=qgh+r$ かつ $d(r)<d(gh)$ である。従って，$f=(qh)g+r$ かつ $d^*(r)\leqq d(r)<d(gh)=d^*(g)$ である。 □

系 4.38. R を任意の可換環とし，$\langle X, <\rangle$ を任意の整列集合とする。R から X へのユークリッド関数が存在するとき，ユークリッド関数での全体を D とし，関数 $d_0 : R \to X$ を

$$d_0(x) = \min\{d(x) \mid d \in D\}$$

と定義すると d_0 は (2) を満足するユークリッド関数である。この d_0 を**最小ユークリッド関数** (the least Eucledean function) と呼ぶ。

証明: a，$b \neq 0$ を任意に取り，$d_0(b) = d(b)$ を実現する $d \in D$ を選ぶと，d はユークリッド関数だから，q, r が存在して $a = qb + r$ かつ $d(r) < d(b)$ である。当然 $d_0(r) \leqq d(r) < d(b) = d_0(b)$ だから，d_0 自身ユークリッド関数である。もし，d_0 が (2) を満たさなければ，前定理により，(2) を満たすユークリッド関数で d_0 よりも小さいものが存在するが，それは d_0 の最小性に反する。 □

問題 4.39. d_0 を可換環 R から $X = \mathbb{N} \cup \{-\infty\}$ への最小ユークリッド関数とする。$d_0(0) = -\infty$，$R^\times = d_0^{-1}(0)$ を示せ。

問題 4.40. $R = \mathbb{Z}$，$X = \mathbb{N} \cup \{-\infty\}$ とするとき，最小ユークリッド関数を求めよ。

4.6 GCD の一意性について★

GCD が唯一つに定まらないことは，数学的見地からは，特に困ることではない。しかし，アルゴリズムをコンピュータ上に実装し，それを利用する上では，アルゴリズムが答えとして返すものがなんであるかが明確に定まっていたほうが都合がいい。

先に述べたように，$u \in R$ を単元とすると，a が f と g の GCD ならば，ua もまた GCD である。$R = K[x]$ の場合，a が GCD ならば，モニックな多項式 $a/\mathrm{lc}(a)$ も GCD であり，a よりも好都合な性質を持つことが多い。そこで，アルゴリズムからの出力の GCD としては，常にモニックな多項式を返すことが考えられる。

この考えを一般の整域 R で有効にするために，正規形という概念を導入する。

定義 4.41 (正規形)**.**
整域 R の各元 $a \in R$ に対して，$a \sim \mathrm{normal}(a)$ であるような要素 $\mathrm{normal}(a)$ が選ばれているものとする。これを a の**正規化**と呼ぶ。また，$a = \mathrm{normal}(a)$ のとき $a \in R$ は**正規形**をしているという。当然 $\mathrm{normal}(0) = 0$ だから 0 は正規形の要素である。さらに次の 2 つの性質を要請する。

- $a \sim b \iff \mathrm{normal}(a) = \mathrm{normal}(b)$
- $\mathrm{normal}(a) \cdot \mathrm{normal}(b) = \mathrm{normal}(a \cdot b)$

このとき R を**正規形が定義された**整域と呼ぶ。

問題 4.42. 正規形が定義された整域を R とする。0 でない $a \in R$ に対して $a = u \cdot \mathrm{normal}(a)$ となる単元 u が唯一つ定まることを示せ。この u を a の**頭単元**（leading unit）と呼び，$\mathrm{lu}(a)$ と記す。一般に $\mathrm{lu}(a)\mathrm{lu}(b) = \mathrm{lu}(ab)$ を示せ。単元の正規形は 1 であることを示せ。

問題 4.43. $\mathbb{Z}$ に正規形を定義すると必然的に $\mathrm{normal}(a) = |a|$ となり，$K[x]$ に正規形を定義すると必然的に $\mathrm{normal}(a) = a/\mathrm{lc}(a)$ となることを示せ。

問題 4.44. R を正規形が定義された整域とする。$R[x]$ 上で $\mathrm{normal}(f) = f/\mathrm{lu}(\mathrm{lc}(f))$ と定義するとき $\mathrm{normal}(f)$ は $R[x]$ の正規形を定めることを示せ。

問題 4.45. $\mathbb{Z}[\mathrm{i}]$ に正規形が定義されているとする。このとき，normal(4) は定義によらず，唯一つに定まる。その値を求めよ。

R を正規形が定義された整域とするとき，記法 $\gcd\{a_1,\ldots,a_n\}$ は $a_1,\ldots,a_n$ の GCD のうち正規形の要素を意味するものとする。$\gcd\{a_1,\ldots,a_n\}$ は当然唯一つに定まる。同様に記法 $\mathrm{lcm}\{a_1,\ldots,a_n\}$ は $a_1,\ldots,a_n$ の LCM のうち正規形の要素を意味するものとする。

正規形が定義された一般のユークリッド整域 R 上で，$f,g\in R$ に対して，この意味での $r=\gcd\{f,g\}$ と，$fs+gt=r$ となるような $s,t\in R$ を求めるアルゴリズムは，次のようになる。

正規形の GCD を求める拡張ユークリッド互除法（Extended Euclidean Algorithm）

入力 $f,g\in R$
出力 $r=\gcd\{f,g\}$ と $fs+gt=r$ となる $s,t\in R$

(1) $\rho_0 \longleftarrow \mathrm{lu}(f)$; $r_0 \longleftarrow f/\rho_0$; $s_0 \longleftarrow 1/\rho_0$; $t_0 \longleftarrow 0$;
$\rho_1 \longleftarrow \mathrm{lu}(g)$; $r_1 \longleftarrow g/\rho_1$; $s_1 \longleftarrow 0$; $t_1 \longleftarrow 1/\rho_1$; $i \longleftarrow 1$;
(2) **while** $r_i \neq 0$ **do**
$q_i \longleftarrow r_{i-1} \div r_i$; $\rho_{i+1} \longleftarrow \mathrm{lu}(r_{i-1}-q_i r_i)$;
$r_{i+1} \longleftarrow (r_{i-1}-q_i r_i)/\rho_{i+1}$; $s_{i+1} \longleftarrow (s_{i-1}-q_i s_i)/\rho_{i+1}$; $t_{i+1} \longleftarrow (t_{i-1}-q_i t_i)/\rho_{i+1}$;
$i \longleftarrow i+1$;
(3) $r \longleftarrow r_{i-1}$; $s \longleftarrow s_{i-1}$; $t \longleftarrow t_{i-1}$;

例 4.46. 例えば $R=\mathbb{Q}[x]$，$f=18x^3-42x^2+30x-6$，$g=-12x^2+10x-2$ の場合は，

i	q_i	ρ_i	r_i	s_i	t_i
0		18	$x^3-(7/3)x^2+(5/3)x-(1/3)$	$1/18$	0
1	$x-3/2$	-12	$x^2-(5/6)x+1/6$	0	$-1/12$
2	$x-1/2$	$1/4$	$x-1/3$	$2/9$	$(1/3)x-1/2$
3		1	0	$-(2/9)x+1/9$	$-(1/3)x^2+(2/3)x-1/3$

と計算が進み，r_i は常にモニックな多項式となる。従って $r=r_2=x-\dfrac{1}{3}$ もモニックになる。それにあわせて $s_2=\dfrac{2}{9}$，$t_2=\dfrac{1}{3}x-\dfrac{1}{2}$ も調整され，

$$x+\frac{1}{3}=(18x^3-42x^2+30x-6)\cdot\frac{2}{9}+(-12x^2+10x-2)\left(\frac{1}{3}x-\frac{1}{2}\right)$$

という関係は保たれる。

問題 4.25 と同じ性質が，ρ_i を考慮に入れて若干修正することで，このアルゴリズムについても成立する。

問題 4.47. アルゴリズム内の変数 ρ_i, r_i, t_i, s_i を用いて T_i を問題 4.25 と同様に，また Q_i は，

$$Q_i \stackrel{\text{def}}{=} \begin{pmatrix} 0 & 1/\rho_{i+1} \\ 1 & -q_i/\rho_{i+1} \end{pmatrix}$$

と定義する。このとき次を示せ。

(a) $(f\ g)\begin{pmatrix} s_i \\ t_i \end{pmatrix} = r_i$，従って $(f\ g)T_i = (r_i\ r_{i+1})$ である。
(b) $T_i = T_0 Q_1 \cdots Q_i$
(c) (任意の i について) $r = \gcd\{r_i, r_{i+1}\}$，つまり r は r_i と r_{i+1} の正規形の GCD である。従って，特に $r = \gcd\{r_0, r_1\} = \gcd\{f, g\}$ である。
(d) $\det T_i = s_i t_{i+1} - t_i s_{i+1} = (-1)^i/(\rho_0 \cdots \rho_{i+1})$
(e) $f = (-1)^i \rho_0 \cdots \rho_{i+1}(t_{i+1} r_i - t_i r_{i+1})$，$g = (-1)^i \rho_0 \cdots \rho_{i+1}(-s_{i+1} r_i + s_i r_{i+1})$

4.7 法逆元計算（Modular inverse）

法計算とは，一般には，可換環 R をイデアル $I \subset R$ で割った剰余環 R/I において，和や積の計算をすることといえる。しかし，狭義にはユークリッド整域の中で (さらに狭義には整数環 $\mathbb{Z}$ の中で)，ある要素で割った余りを用いて計算をすることを言う。

定理 4.48. (R, d) をユークリッド整域とする。$a, m \in R$ に対し，$a \bmod m \in R/mR$ が単元である，すなわち逆元を持つための必要十分条件は $\gcd\{a, m\} = 1$ である。この場合，拡張互除法を応用して，R/mR における $a \bmod m$ の逆元を計算できる。

証明:

$$\exists s \in R\ sa \equiv 1 \pmod{m} \iff \exists s, t \in R\ sa + tm = 1 \Longrightarrow \gcd\{a, m\} = 1$$

である。逆に $\gcd\{a, m\} = 1$ なら，拡張互除法を応用して，$sa + bm = 1$ となる s, t が計算できる。このとき $s \bmod m$ は $a \bmod m$ の逆元である。 □

すなわち $\langle r, s, t\rangle \longleftarrow \mathrm{EEA}(a, m)$ を計算した結果，$r = 1$ であれば，自動的に $s \bmod m$ が R/mR における $a \bmod m$ の逆元となる。$r \neq 1$ ならば，$a \bmod m$ は R/mR において単元ではなく，その逆元は存在しない。

例 4.49. $R = \mathbb{Z}$，$m = 29$，$a = 12$ とする。拡張互除法によれば，

i	q_i	r_i	s_i	t_i
0		29	1	0
1	2	12	0	1
2	2	5	1	-2
3	2	2	-2	5
4	2	1	5	-12
5		0	-12	29

のように計算が進行し，$5 \cdot 29 + (-12)12 = 1$ が得られる。よって $(-12) \cdot 12 \equiv 17 \cdot 12 \equiv 1 \pmod{29}$，つまり $17(\equiv -12)$ が 29 を法とする 12 の逆元である。 □

例 4.50. $R=\mathbb{Q}[x]$，$m=x^3-x+2$，$a=x^2$ とする。拡張互除法によれば

i	q_i	r_i	s_i	t_i
0		x^3-x+2	1	0
1	x	x^2	0	1
2	$-x-2$	$-x+2$	1	$-x$
3	2	4	$x+2$	$-x^2-2x+1$

のように計算が進行し，$(x+2)(x^3-x+2)+(-x^2-2x+1)x^2=4$ が得られる。よって $(-x^2-2x+1)x^2\equiv 4 \pmod{x^3-x+2}$，すなわち $(-x^2-2x+1)/4$ が x^3-x+2 を法とする x^2 の逆元である。 □

$p\in\mathbb{Z}$ が素数であったり，$f\in K[x]$ が既約であったりすれば，$\mathbb{Z}/p\mathbb{Z}$ や $K[x]/\langle f\rangle$ が体になることが，定理 4.48 より得られる。

問題 4.51. 次の計算をせよ。$28^{-1} \bmod 75$

問題 4.52. $7785m\equiv 16 \pmod{8477}$ であるような整数 m を 1 つ求めよ。

問題 4.53. a,b を $a\perp b$ なる正の整数とする。系 3.18 より，任意の整数 $r\in\mathbb{Z}$ が整数 $s,t\in\mathbb{Z}$ により $r=sa+tb$ と書けるが，非負整数 $s,t\in\mathbb{N}$ では $r=sa+tb$ と書けない整数 r の最大値は何か？

解答例 $ab-a-b$ である。

実際，$ab-a-b=sa+tb$ とすると，$a\perp b$ から簡単な計算により，$s\equiv -1 \bmod b$，$t\equiv -1 \bmod a$ だが，$s=ib-1$，$t=ja-1$ とすると，$sa+tb=(i+j)ab-a-b$ となり，$1=i+j$ が得られる。よって，i，j のうち一方は 0 以下でなければならず，s，t の一方は負である。

$ab-a-b<r=sa+tb$ とする。$(s\pm b)a+(t\mp a)b=sa+tb$ だから，適当に増減すれば s の値を $[0..b)$ の範囲に収めることができる。このとき $t\geq 0$ である（そうでなければ $sa+tb\leq (b-1)a-b=r$ となり，仮定に反する）。よって，s,t はともに非負整数にとることができる。 □

問題 4.54. 52 円と 21 円の切手がたくさんある。

(a) ちょうど 1000 円分の切手を貼るにはこれらを何枚ずつ使えば良いか。
(b) ちょうど 1600 円分の切手を貼るにはこれらを何枚ずつ使えば良いか。
(c) これらを使って貼ることのできない最高額はいくらか。

解答例 (a) $52m+21n=1000$ とおくと，$21n\equiv -40 \bmod 52$ である。52 を法とする 21 の逆数を（拡張互除法で）求めると 5 だから，$n\equiv -40\times 5\equiv 8 \bmod 52$ である。$n=8$ とおいてみると $m=16$ が得られる。従って 52 円切手を 16 枚，21 円切手を 8 枚使えばよい。

(b) (a) と同様である。$21n\equiv 1600\equiv 40 \bmod 52$ を解くと，$n\equiv 44 \bmod 52$ である。$n=44$ とおいてみると，$m=13$ が得られる。従って 52 円切手を 13 枚，21 円切手を 44 枚使えばよい。

(c) 前問より $(52\times 21-52-21=)$ 1019 円 □

問題 4.55. 方程式 $x^3-x^2-x-1=0$ の根の 1 つを α とする．$\alpha^2+\alpha$ の逆数を $A\alpha^2+B\alpha+C$ の形に表わせ．ただし，A,B,C は有理数とする．

解答例　多項式環 $\mathbb{Q}[x]$ で $r_0(x) = x^3 - x^2 - x - 1$，$r_1(x) = x^2 + x$ として，拡張ユークリッド互除法を行なう．

i	q_i	r_i	s_i	t_i
0		$x^3 - x^2 - x - 1$	1	0
1	$x - 2$	$x^2 + x$	0	1
2	$x + 2$	$x - 1$	1	$-x + 2$
3		2	$-x - 2$	$x^2 - 3$

により，$2 = (x^3 - x^2 - x - 1)(-x - 2) + (x^2 + x)(x^2 - 3)$ である．したがって

$$2 = (\alpha^3 - \alpha^2 - \alpha - 1)(-\alpha - 2) + (\alpha^2 + \alpha)(\alpha^2 - 3) = (\alpha^2 + \alpha)(\alpha^2 - 3)$$

だから $\dfrac{1}{(\alpha^2 + \alpha)} = \dfrac{1}{2}\alpha^2 + 0\alpha - \dfrac{3}{2}$ である． □

問題 4.56. 定数 $\alpha \in K$ と多項式 $g(x) \in K[x]$ を考える。$g(\alpha) \neq 0$ とするとき，$g(x)$ を法とする $x - \alpha$ の逆元は

$$p(x) = \frac{-1}{g(\alpha)} \left(\frac{g(x) - g(\alpha)}{x - \alpha} \right)$$

とかけることを証明せよ。

解答例　$p(x) \in K[x]$ であることは，因数定理により，$g(x) - g(\alpha)$ が $x - \alpha$ で割り切れ，$g(\alpha) \neq 0$ であることよりわかる。また，$(x - \alpha)p(x) \equiv 1 \pmod{g(x)}$ であることは簡単な計算でわかる。 □

一般に有限体は要素数が同じならば同型になることが知られており，要素数 p の有限体は $\mathrm{GF}(p)$ と書かれる。従って，p が素数のとき，剰余環 $\mathbb{Z}/p\mathbb{Z}$ も単に $\mathrm{GF}(p)$ と表記する。さらに，有限体の要素数は，必ず素数 p の冪 $q = p^n$ になることが知られており，$f \in \mathrm{GF}(p)[x] = (\mathbb{Z}/p\mathbb{Z})[x]$ が n 次の既約式ならば，$\mathrm{GF}(p)[x]/\langle f \rangle$ の要素数は p^n になる。従って，$\mathrm{GF}(p)[x]/\langle f \rangle$ の構造は，既約式 f の次数だけによって定まり，特に f を明示する必要がなければ，$\mathrm{GF}(p^n)$ と書ける。

補題 4.57. K を体とし，$f \in K[x]$ をモニックな既約多項式で定数ではないものとする。このとき $K[x]/\langle f \rangle$ は K の拡大体で，$\alpha = (x \bmod f) \in K[x]/\langle f \rangle$ とすると，α は f のゼロ点である。 □

例 4.58. $R = \mathrm{GF}(5)[x]$，$f = x^3 - x + 2$ とする。f は $\mathrm{GF}(5)$ 内にゼロ点を持たないので既約である。f は 3 次だから，$\mathrm{GF}(5)[x]/\langle f \rangle = \mathrm{GF}(5^3) = \mathrm{GF}(125)$ である。$g = x^2$ として拡張互除法を行なうと

$$(-x - 2)(x^3 - x + 2) + (x^2 + 2x - 1)x^2 = 1$$

が得られる。よって $x^2 + 2x - 1 \bmod f$ が f を法とする x^2 の逆元である。$x \bmod f$ を α と略記するなら，$f(x) \bmod f = 0$，すなわち $f(\alpha) = 0$ だから，α は f のゼロ点になり，$\alpha^2 + 2\alpha - 1 = (\alpha^2)^{-1} = \alpha^{-2} \in \mathrm{GF}(125)$ である。 □

定理 4.33, 4.34 の系として次が得られる

系 4.59. K を体とし，$f \in K[x]$ を n 次とする。$K[x]/\langle f \rangle$ での加減乗算と可逆元による除算は，体 K 上での演算 $O(n^2)$ 回でできる。

系 4.60. $m \in \mathbb{Z}$ で，$\lambda(m) = n$ とする。$\mathbb{Z}/m\mathbb{Z}$ での加減乗算と可逆元による除算は，単精度演算 $O(n^2)$ 回でできる。

ユークリッド環 R 上の法計算による冪は，次のアルゴリズム power により比較的容易に計算できる。この手法を反復 2 乗法 (Repeated squaring) と呼ぶ。

反復 2 乗法

入力 $a, m \in R$, $n \in \mathbb{N}$
出力 $a^n \bmod m \in R$
power(a, n, m)
  if $n = 0$ **then return** 1;
  $(p, k) \longleftarrow n \div 2$;
  $b \longleftarrow$ power(a, p, m);
  if $k = 0$ **then return** $b \cdot b \bmod m$ **else return** $b \cdot b \cdot a \bmod m$;

上のアルゴリズムは $\lambda(n)$ に比例する回数の R 上での法乗算により実行できる。例えば $8^{13} \bmod 17$ を計算するには，$\mathbb{Z}$ 上で $8^{13} = 549755813888$ を計算してから 17 による剰余を求めるより，$8^2, 8^3, 8^6, 8^{12}, 8^{13}$ の 17 による剰余を各段階で求めて行くほうが容易である。

問題 4.61. 2^{1000} の 10 進末尾 4 桁を計算せよ。

反覆 2 乗法を使えば，オイラーの定理 (定理 3.81) またはフェルマーの小定理 (定理 3.82) を利用しても $\mathbb{Z}$ 上の法逆元計算を効率よく計算できる。すなわち，p が素数ならば，任意の $m \in \{1, 2, \ldots, p-1\}$ に対して

$$1 \equiv m^{\varphi(p)} = m^{p-1} = m \cdot m^{p-2} \pmod{p}$$

より，$m^{p-2} \bmod p$ を反覆 2 乗法で計算すればよい。

しかし，この計算法の計算量は，特に工夫をしない限り $O(\lambda(m)^3)$ だから，先の拡張互除法を使う場合にかなわないし，拡張互除法にはこれから述べるような応用が他にもたくさんある。

4.8 中国剰余算法（Chinese remainder algorithm）

法逆元計算の第一応用は中国剰余定理の計算版である。中国剰余定理 (定理 3.58) や中国剰余写像 (定理 3.68) は $\mathbb{Z}$ の場合から一般のユークリッド整域の場合に容易に拡張できるので，その拡張をした上で，計算版を与えよう。

R をユークリッド整域とし，$n_1, \ldots, n_r \in R$ を対ごとに互いな素な要素とする。$n = n_1 \cdots n_r$ とすると $n = \mathrm{lcm}\{n_1, \ldots, n_r\}$ でもある。$R/\langle n \rangle$ から直積 $\prod_{i=1}^r R/\langle n_i \rangle$ への写像 θ を

$$\theta(a \bmod n) \stackrel{\mathrm{def}}{=} \langle a \bmod n_1, \ldots, a \bmod n_r \rangle$$

により定義すると，θ は well-defined あり，同型写像になる。特に乗法群に限った場合でも，$(R/\langle n \rangle)^\times \cong \prod_{i=1}^r (R/\langle n_i \rangle)^\times$ が成り立つ。

定理 4.62 (中国剰余定理—計算版)**.**
上記のような $n_i \in R$ $(i = 1, 2, \ldots, r)$ と $a_i \bmod n_i$ $(i = 1, 2, \ldots, r)$ とが与えられたとする。

$$n := \prod_{i=1}^k n_i, \quad n_i^* := n/n_i, \quad b_i := (n_i^*)^{-1} \bmod n_i, \quad a := \sum_{i=1}^n a_i n_i^* b_i$$

とすると，$a \equiv a_i \pmod{n_i}$ $(i = 1, 2, \ldots, n)$ である。

問題 4.63. 上記の定理 (中国剰余定理—計算版) を証明せよ。

問題 4.64. 上記の定理に基づいて実際に $a \equiv a_i \pmod{n_i}$ $(i = 1, 2, \ldots, n)$ かつ $d(a) < d(n)$ を満たす a を求めるアルゴリズムを与えよ。

系 4.65.
$f_1, \ldots, f_r \in K[x]$ を互いに素な多項式で，$f = f_1 \cdots f_r$ とする。$d_i = \deg f_i \geqq 1$，$n = \deg f = \sum d_i$ とするとき，$a_i \in K[x]$ $(\deg a_i < d_i)$ に対し

$$a \equiv a_i \pmod{f_i}, \quad \deg f < n$$

となる唯一の $a \in K[x]$ は K 上の演算 $O(n^2)$ 回で計算することができる。

系 4.66.
$m_1, \ldots, m_r \in \mathbb{Z}$ を互いに素な正の整数で $m = m_1 \cdots m_r$ とする。$n = \mathrm{len}(m)$ とするとき，$a_i \in \mathbb{Z}$ $(0 \leq a_i < m_i)$ に対し

$$a \equiv a_i \pmod{m_i}, \quad 0 \leq a < m$$

となる唯一の $a \in \mathbb{Z}$ は単精度演算 $O(n^2)$ 回で計算することができる。

中国剰余の計算は逐次的に行うこともできる。次の 2 つの問題でそれを扱う。

問題 4.67. R をユークリッド整域，$n_1, n_2 \in R$ は $n_1 \perp n_2$ を満たすとするとき，任意の $a_1, a_2 \in R$ に対して

$$b := n_1^{-1} \bmod n_2, \quad c := (a_2 - a_1)b \bmod n_2, \quad a := a_1 + n_1 c, \quad n := n_1 n_2$$

とすると，$a \equiv a_i \pmod{n_i}$ $(i = 1, 2)$ である。

問題 4.68. 上の問題を利用して，対ごとに互いに素な R の要素 $n_1, n_2, \ldots, n_k \in R$ と，任意の $a_1, a_2, \ldots, a_k \in R$ とに対して中国剰余定理の a と n を逐次的に計算するアルゴリズムを与えよ。

問題 4.69. n を正の整数とする。$\alpha_1, \ldots, \alpha_k \in (\mathbb{Z}/n\mathbb{Z})^\times$ に対して，$\alpha_1^{-1}, \ldots, \alpha_k^{-1}$ をまとめて計算するアルゴリズムを与えよ。ただし，その過程において，n を法とする逆元はある数 a に対して一回計算するだけにとどめ，法乗算も $3k$ 回以下にとどめよ。一般に，法逆元の計算は，定数倍ではあるが法乗算よりも時間がかかるので，この結果は，実際の計算において有用である。

4.9 行列積の法計算 (Modular Matrix Multiplication)

前節の「計算版」中国剰余定理 (定理 4.62) の応用は，限りがないが，その第一のものは多精度計算の高速化である。ここでは，まず，行列の積の計算を通じて，その威力を見てみる。

2 つの m 次正方行列 A と B の積の計算 $C = AB$ を考えると，C の各成分は

$$C[i, j] = \sum_{k=1}^{m} A[i, k]B[k, j]$$

として計算できるが，この計算を素朴に行うと m 回の乗算を要する。今，A，B の成分が l 桁の整数とすると，この計算には $O(l^2m)$ の時間がかかる。従って C の全成分を計算には $O(l^2m^3)$ の時間がかかると考えられる。

そこで，上の式で直接に $C[i,j]$ の計算をせずに，中国剰余定理を利用することを考える。行列 A の各成分を n で割った余りを成分とする行列を $A \bmod n$ と記す。このとき $C \mod n = AB \mod n$ の成分は

$$(C \bmod n)[i,j] = \left(\sum_{k=1}^{m}(A \bmod n)[i,k](B \bmod n)[k,j]\right) \bmod n$$

である。n が十分に小さい数であれば，この計算に要する時間は $O(m)$ だから，$C \bmod n$ の全成分を計算しても，その計算量は $o(m^3)$ である。

さて，今，A，B の成分は，どれも絶対値が M 以下の整数とすると，$C = AB$ の成分の絶対値は高々 mM^2 である。そこで $n = \prod_{i=1}^{k} n_i > 2mM^2$ となるように，対ごとに互いに素な正の整数を $n_1, n_2, \ldots, n_k$ を適当にとり，$C \bmod n_i$ を考える。中国剰余定理により，$(C \bmod n_i)[r,s] \equiv d \pmod{n_i}$ $(i = 1, 2, \ldots, k)$ なる d は，各成分 $[r,s]$ ごとに $-n/2 < d < n/2$ の範囲で一意に決まるから，そのような d は $C[r,s]$ に他ならない。

行列の積を計算するアルゴリズムは，上の考察に沿った法計算を用いると，次のようになる。

法計算による行列積

入力　多精度整数を成分とする m 次正方行列 A, B (ただし，各成分の絶対値は M 以下)
出力　行列の積 $C = AB$

(1) 対ごとに互いに素な正の整数 $n_1, n_2, \ldots, n_k$ を十分にとり，
　　$\prod_{i=1}^{k} n_i > 2mM^2$ となるようにする。
(2) **for** $i \in [1..k]$ **do**
　　$A_i \longleftarrow A \bmod n_i; B_i \longleftarrow B \bmod n_i;$
　　for $r \in [1..m]$ **do for** $s \in [1..m]$ **do** $C_i[r,s] \longleftarrow (\sum_{t=1}^{m} A_i[r,t]B_i[t,s]) \bmod n_i;$
(3) **for** $r \in [1..m]$ **do for** $s \in [1..m]$ **do**
　　$C[r,s] \longleftarrow \mathrm{ChineseRemainder}(C_1[r,s] \bmod n_1, \ldots, C_k[r,s] \bmod n_k);$

今，M が l 桁の数とし，m も l に比べて極端に大きくないものとすれば，n_i たちをその法演算が単位時間におさまるように十分小さくとっても，$k = O(l)$ である。n_i たちはあらかじめ計算しておいたものを必要な数だけ使えば良いので，(1) に要する計算量は無視できる。(2) は，各 A_i, B_i の計算に $O(lm^2)$，C_i の計算に $O(m^3)$ の時間を要するから，全 i についてそれらを計算すると $O(l^2m^2 + lm^3)$ の時間を要する。最後に (3) の各 $C[r,s]$ の計算は $O(l^2)$ だから，全 r, s についての計算は $O(l^2m^2)$ となり，アルゴリズム全体の計算量は $O(l^2m^2 + lm^3)$ となる。

例 4.70. 行列計算

$$AB = \begin{pmatrix} 9 & 7 & -6 \\ -8 & -3 & 10 \\ 8 & -10 & 7 \end{pmatrix} \begin{pmatrix} -7 & 3 & 11 \\ -12 & 9 & -6 \\ 8 & -3 & 9 \end{pmatrix} = \begin{pmatrix} -195 & 108 & 3 \\ 172 & -81 & 20 \\ 120 & -87 & 211 \end{pmatrix} = C$$

を考える。$M = 12$，$m = 3$ とすると，$n > 2 \cdot 3 \cdot 23^2 = 864$ ならば良いので $n = 9 \cdot 10 \cdot 11 = 990$ で十分である。すると

$$A \bmod 9 = \begin{pmatrix} 0 & -2 & 3 \\ 1 & -3 & 1 \\ -1 & -1 & -2 \end{pmatrix}, \quad B \bmod 9 = \begin{pmatrix} 2 & 3 & 2 \\ -3 & 0 & 3 \\ -1 & -3 & 0 \end{pmatrix},$$

より，

$$AB \bmod 9 = \begin{pmatrix} 3 & 9 & -6 \\ 10 & 0 & -7 \\ 3 & 3 & -5 \end{pmatrix} \bmod 9 = \begin{pmatrix} 3 & 0 & 3 \\ 1 & 0 & 2 \\ 3 & 3 & 4 \end{pmatrix}$$

が得られる。同様に

$$AB \bmod 10 = \begin{pmatrix} 5 & 8 & 3 \\ 2 & -1 & 0 \\ 0 & 3 & 1 \end{pmatrix}, \quad AB \bmod 11 = \begin{pmatrix} -3 & -2 & 3 \\ -4 & -4 & -2 \\ -1 & 1 & 2 \end{pmatrix}$$

であるから，中国剰余定理によって $(-395..495)$ 内に各成分がおさまるように復元すると C が得られる。

問題 4.71. 大きな係数を持つ 2 つの多項式 $f, g \in \mathbb{Z}[x]$ の積を計算するのに，上のように中国剰余定理を応用することについて論ぜよ。

4.10 行列式の法計算 (Modular Determinant Computation)

A を整数成分からなる行列とする。線形代数の理論によれば，A が正則なら，A の逆行列 A^{-1} は $\tilde{A}/\det A$ で与えられる。ここで $\tilde{A}$ は余因子行列を表す。A^{-1} を直接求めるにしても，$\det A$ を求めるにしても，ガウスの消去法が，効率の良いアルゴリズムとして知られているので，行列の逆行列を求める問題は，結局, 行列式をガウス消去法で求める問題に帰着されると言っていい。

さて，$n \times n$ 行列 A の行列式を $\det A$ をガウスの消去法で具体的に求める手順は，例えば，次のようになる。$A[1,1] \neq 0$ とすると，$i \in [2..n], j \in [2..n]$ に対して，

$$B[i-1, j-1] = A[i,j] - A[i,1]A[1,j]/A[1,1]$$

と置く。後は $\det B$ の値を再帰的に求め，$\det A = A[1,1] \det B$ とすればいい。

$A[1,1] = 0$ の場合は，$A[1,i] \neq 0$ なる i を探し，第 1 行と第 i 行を入れ変えて同じ計算を行い，結果を -1 倍すればいい。そのような i が存在しなければ，$\det A = 0$ である。

多くの数値計算においては，浮動小数点表記を用いて上記の計算を行なえば良いので，これ以上工夫しなくとも，$O(n^3)$ 回の演算で行列式の値が求まる。しかし，浮動小数点表記は，誤差が出るので，代数的な応用では利用できないことも多い。その場合，分子と分母を多精度整数とする必要があるが，そのような数にガウスの消去法を使っても，結局，途中の計算結果の桁数が分子分母とも膨大になるため，浮動小数点を用いたときのような効率が得られない。

このような場合，比較的小さな素数をたくさん用いた法計算が有効である。

法行列式計算

入力 行列 $A \in \mathbb{Z}^{n\times n}$，ただし $\forall i,j\ |A[i,j]| < B$ とする。
出力 $\det A \in \mathbb{Z}$

determinant(A)
(1) $m_1 \cdots m_r > 2n^{n/2}B^n$ となるように r 個の素数 $m_1,\ldots,m_r \in \mathbb{N}$ を選ぶ;
(2) **for** $i \in [1..r]$ **do**
(3) ガウスの消去法により $\mathbb{Z}/m_i\mathbb{Z}$ 上で $d_i = (\det A) \bmod m_i = \det(A \bmod m_i)$ を計算する;
(4) $a \bmod m \longleftarrow \mathrm{ChineseRemainder}(d_1 \bmod m_1,\ldots,d_r \bmod m_r)$;
(5) **return** a; (ただし $|a| < m/2$)

例 4.72. $A = \begin{pmatrix} 4 & 5 \\ 6 & -7 \end{pmatrix}$ とする。$2\cdot3\cdot5\cdot7 = 210 > 98 = 2\cdot2^1\cdot7^2$ だから，最初の 4 つの素数 $2,3,5,7$ を選ぶと

$$\det A \equiv \det\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \equiv 0 \pmod 2$$
$$\det A \equiv \det\begin{pmatrix} 1 & -1 \\ 0 & -1 \end{pmatrix} \equiv -1 \pmod 3$$
$$\det A \equiv \det\begin{pmatrix} -1 & 0 \\ 1 & -2 \end{pmatrix} \equiv 2 \pmod 5$$
$$\det A \equiv \det\begin{pmatrix} -3 & -2 \\ -1 & 0 \end{pmatrix} \equiv -2 \pmod 7$$

だから，中国剰余算法で求めると $\det A \equiv 362 \pmod{210}$ だが，$|\det A| < 105$ より $\det A = -58$ である。

定理 4.73.
$n\times n$ 行列 $A \in \mathbb{Z}^{n\times n}$ のどの成分の絶対値も B 未満とする。このとき $\det A$ は $\tilde{O}(n^4 \log B + n^3(\log B)^2)$ で計算できる。

4.11 フェルマの 2 平方定理—計算版

フェルマの 2 平方定理 (定理 3.116) によれば，$4n+1$ 型の素数は整数 r,t によって r^2+t^2 の形に書ける。拡張互除法の別の応用として，そのような素数 p が与えられたとき，この r,t を効率よく計算する方法について論ずる。

ここで $\mathrm{EEA}(f,g)$ は 4.3 節の拡張互除法が入力 f,g に対して計算する列 $\{(r_i,s_i,t_i)\}$ $(i = 1,2,\ldots,\lambda+1)$ を表すものとする。$r_\lambda = \gcd\{f,g\}$ であり，$r_{\lambda+1} = 0$ とする。

定理 4.74 (Thue の補題—計算版)**.** $n,b,R,T \in \mathbb{Z}$ が $0 \leqq b < n$ と $0 < R \leqq n < RT$ を満たすものとする。このとき $\mathrm{EEA}(n,b) = \{(r_i,s_i,t_i)\}$ とし，j を $r_j < R$ を満たす最小の j とすると，次が成り立つ。

$$r_j \equiv bt_j \pmod n, \quad 0 \leqq r_j < R, \quad 0 < |t_j| < T$$

証明: $r_0 = n \geqq R > 0 = r_{\lambda+1}$ だから，定理の要件を満たす j は，存在し 1 つに定まる。条件より，もちろん $j \geqq 1$ かつ $r_{j-1} \geqq R$ である。すると問題 4.32 の (b) と定理の条件より

$$|t_j| \leqq n/r_{j-1} \leqq n/R < T$$

である。問題 4.32 の (a) より，$|t_j| \geqq |t_1| > 0$ であり，$r_j = ns_j + bt_j$ だから $r_j \equiv bt_j \pmod{n}$ である。 □

上の定理を踏まえて，フェルマの 2 平方定理 (定理 3.116) の証明を再吟味すれば，次の定理が得られる。

定理 4.75 (フェルマの 2 平方定理—計算版)**.** p は素数であり 4 を法として 1 と合同とする。$b^2 \equiv -1 \pmod{p}$ とし, $\mathrm{EEA}(p, b) = \{(r_i, s_i, t_i)\}$ とする。このとき，j を $r_j^2 < p$ を満たす最小の j とすると，$p = r_j^2 + t_j^2$ である。

この定理より，$4n + 1$ 形の素数を 2 つの平方数の和に分解する効率的なアルゴリズムが得られる。

平方数の和への分解

入力 $p \equiv 1 \pmod{4}$ を満たす素数 p
出力 $p = r^2 + t^2$ を満たす整数 $r, t \in \mathbb{Z}$

(1) $p - 1 = s2^e$ なる奇数 s と整数 e を求める (入力条件より $e > 1$ である);
(2) **repeat** $k \in [1..p)$ を選ぶ; $m \longleftarrow k^s \bmod p$; **until** $m \neq 1 \wedge m \neq p - 1$
(3) **repeat** $b \longleftarrow m$; $m \longleftarrow b^2 \bmod p$; **until** $m = p - 1$
(4) $\mathrm{EEA}(b, p) = \{(r_i, s_i, t_i)\}$ を計算し，$r_i^2 < p$ を満たす最小の i を求める;
(5) $r \longleftarrow r_i$; $t \longleftarrow t_i$;

問題 4.76. 上のアルゴリズムにおいて (1)–(3) は，p を法とする -1 の平方根 b を探すステップである。(1)–(3) によって，このような b が見つかることを証明せよ。

例 4.77. $p = 1009 \equiv 1 \pmod{4}$ であり，p は素数である。$p - 1 = 1008 = 63 \times 2^4$ だから，(1) では $s = 63$，$e = 4$ である。(2) で $k = 2$ として m を計算すると $m = 192$ となり **until** の条件を満たす。(3) では $b = 540$ のとき，$m = 1008 = p - 1$ となり **until** の条件を満たす。(4) の拡張互除法により次のような列が得られる。

i	q_i	r_i	s_i	t_i
0		1009	1	0
1	1	540	0	1
2	1	469	1	-1
3	1	71	-1	2
4	6	43	7	-13
5	1	28	-8	15
6	1	15	15	-28
⋮	⋮	⋮	⋮	⋮

${r_5}^2 = 28^2 < 1009 \leqq 43^2 = {r_4}^2$ だから，$r = r_5 = 28$，$t = t_5 = 15$ となるが，実際 $1009 = 28^2 + 15^2$ である。

問題 4.78. 次の素数 p について，p を法とする -1 の平方根を求めよ。

(a) $p = 89$
(b) $p = 1061$
(c) $p = 1877$
(d) $p = 2689$
(e) $p = 3617$

問題 4.79. 次の素数 p について，p を 2 つの平方数の和の形に表せ。

(a) $p = 89$
(b) $p = 1061$
(c) $p = 1877$
(d) $p = 2689$
(e) $p = 3617$

問題 4.80. 次の素数 p を面積に持つ格子点正方形（xy 座標がともに整数である正方形）を作図したい。頂点の 1 つを原点 $(0, 0)$ にとった場合，他の 3 頂点はどこにとればよいか。解は 1 つには定まらないが，その 1 例を挙げよ。

(a) $p = 2689$
(b) $p = 3617$

4.12 分数小数変換

拡張互除法の次の応用として分数小数変換について考える。有理数は，その定義より，互いに素な整数を分子分母とする分数 s/t の形 (既約分数) に書ける。特に，分母が正となるように選ぶと，その表現は一意に定まる。また，任意の実数は，任意の進法で無限小数として書け，有限小数になる場合を除き，この無限小数としての表記は一意である。特に，有理数は，任意の進法で循環小数として表現できる。

さて，特に 10 進表記の場合に，正の有理数の既約分数表記と循環小数表記との間の変換アルゴリズムについて考えよう。本節で扱うアルゴリズムは，何の困難もなく，任意 n 進表記場合に一般化できる。

まず既約分数から循環小数への変換であるが，これは，前に述べた分数を循環 2 進小数に変換する場合 (2.2.4 節) と，同じ考え方でできる。2/7 を例に説明しよう。

$$\begin{array}{cccccccccc} 2/7 & 6/7 & 4/7 & 5/7 & 1/7 & 3/7 & 2/7 & 6/7 & 4/7 & \cdots \\ & 2 & 8 & 5 & 7 & 1 & 4 & 2 & 8 & \cdots \end{array}$$

2.2.4 節と同様，左端の数値が変換したい分数 α である。その右の下の段にはその数値を 10 倍した結果の整数部分 $\lfloor 10\alpha \rfloor$，その上には小数部分 $10\alpha \bmod 1 = 10\alpha - \lfloor 10\alpha \rfloor$ が書かれている。さらに，上の段の数値を 10 倍して，右隣の列の下段にその結果の整数部分を，上段に小数部分を，次々と書き並べて行く。下段には 0–9 の数字が現れるので，その数字を小数点の右に書き並べれば α の 10 進小数表記がえられる。α が有理数ならば，やがて上段の値は繰り返すようになり，当然，下段の数字も繰り返すので，展開を終了して，循環小数として書くことができる。上の例では，$2/7 = 0.\dot{2}85714\dot{}$ である。上の計算は，基本的には，小学校以来の筆算で展開するの同じだからこれ以上の説明は不要だろう。

逆に有理数 α の無限小数表記 $0.d_1d_2\cdots$ が与えられたとき，既約分数 s/t の形に変換するという問題を考えよう。これについても，2 進循環小数表記を分数に変換する場合 (2.2.4 節) と同様の考え方が使える。すなわち，循環節の長さを e とすると，循環節を分子，$10^e - 1$ を分母に持つ分数を考えればよい。たとえば上の $0.\dot{2}85714\dot{}$ の場合，

$$0.\dot{2}85714\dot{} = \frac{285714}{999999} = \frac{2}{7}$$

である。しかし，この方法は，分母が大きい場合には実質的には使えないと言ってよい。分母が t だと，循環節の長さは，通常 $O(t)$ になるからだ。たとえば分母が 4 桁くらいであっても，循環節の長さ何千桁にもなり，そのような巨大な分子分母を持つ分数を扱うのは，たとえ可能だとしても得策とはいえない。

（無限）少数表現から分数表現を復元するもっと現実的なアルゴリズムを考えるために，まず Thue の補題（補題 3.115）について再考する。この補題は任意の n と b について，十分広い範囲を探せば，$r \equiv bt \pmod{n}$ を満たす (r,t) が見つかることを主張するものであったが，この (r,t) の一意性について考えよう。逆に探す範囲があまり広くなければ，このような (r,t) はどれも一定の比を持つようにできる。

定理 4.81. $n, b, R, T \in \mathbb{Z}$ が $0 \leqq R$，$0 < T$，$2RT < n$ を満たすとする。このとき $r, r', t, t' \in \mathbb{Z}$ が

$$\begin{aligned} r &\equiv bt \pmod{n}, \quad |r| \leqq R, \quad 0 < |t| \leqq T \\ r' &\equiv bt' \pmod{n}, \quad |r'| \leqq R, \quad 0 < |t'| \leqq T \end{aligned}$$

を満たせば，$r/t = r'/t'$ である。

証明: $rt' - r't \equiv btt' - bt't \equiv 0 \pmod{n}$ だから

$$|rt' - r't| \leqq |rt'| + |r't| \leqq 2RT < n$$

より $rt' - r't = 0$ である。 □

定理 4.82 (有理数の再構成 (rational reconstruction))**.**
整数 $n, b, R, T \in \mathbb{Z}$ は $0 \leqq b < n$, $0 \leqq R$, $0 < T$ を満たすとする。さらに $\mathrm{EEA}(n, b) = \{(r_i, s_i, t_i)\}$ で, j を $r_j \leqq R$ を満たす最小の j とする。このとき

$$r = ns + bt, \quad |r| \leqq R, \quad 0 < |t| \leqq T \tag{$*$}$$

を満たす $r, s, t \in \mathbb{Z}$ が存在すれば，次が成り立つ。

(a) $0 < |t_j| \leqq T$
(b) $2RT < n$ ならば，0 でない整数 q が存在して $r = r_j q, s = s_j q, t = t_j q$

証明: $r_0 = n \geqq R > 0 = r_{\lambda+1}$ だから，定理の要件を満たす j は，存在し 1 つに定まる。条件より，

$$0 \leqq r_j \leqq R < r_{j-1}, \quad 0 < |t_j|, \tag{A}$$

が成り立ち，定理の条件と EEA の性質より

$$r = ns + bt, \quad r_{j-1} = ns_{j-1} + bt_{j-1}, \quad r_j = ns_j + bt_j \tag{B}$$

である。$\varepsilon = s_j t_{j-1} - s_{j-1} t_j$ とおけば，定理 4.25 の (d) より $\varepsilon = \pm 1$ だから，$\mu = (st_{j-1} - s_{j-1}t)/\varepsilon$, $\nu = (s_j t - st_j)/\varepsilon$ は整数である。

$$s_j \mu + s_{j-1}\nu = s, \quad t_j \mu + t_{j-1}\nu = t \tag{C}$$

であることは，計算で容易に示される。まず $\nu \neq 0$ なら μ と ν が異符号であること，すなわち $\mu\nu < 0$ を示そう。(B) と (C) より

$$r = ns + bt = ns_j\mu + ns_{j-1}\nu + bt_j\mu + bt_{j-1}\nu = r_j\mu + r_{j-1}\nu$$

だから，$\nu \neq 0$ かつ $\mu\nu \geqq 0$ ならば $R \geqq |r| \geqq r_{j-1}$ となり，j の定義に反する。従って $\nu = 0$ か $\mu\nu < 0$ である。

それぞれの場合に (a) を示す。

(1) $\nu = 0$ の場合。(C) より $t_j \mid t$ であり，$t \neq 0$ だから，$|t_j| \leqq |t| \leqq T$ である。
(2) $\mu\nu < 0$ の場合。問題 4.32 の (a) より $t_j t_{j-1} \leqq 0$ だから，$T \geqq |t| = |t_j\mu| + |t_{j-1}\nu| \geqq |t_j|$ である。

次に (b) を示そう。$2RT < n$ とする。(B) より，

$$r_j \equiv bt_j \pmod{n}, \quad r \equiv bt \pmod{n}$$

である。(A)，(a)，(*) と条件 $2RT < n$ を合わせれば，前定理の条件を満たすから，$r/t = r_j/t_j$ である。すると (B) より

$$0 = tr_j - t_j r = tns_j + tbt_j - t_j ns + t_j bt = n(ts_j - t_j s)$$

だから $ts_j = t_j s$ である。定理 4.25 の (d) より $s_j \perp t_j$ だから $t_j | t$ であり，$t = t_j q$ とおくと，$t \neq 0$ だから $q \neq 0$ であり，$r = r_j q$，$s = s_j q$ である。 □

上の定理を少数表現からの分数表現の復元に応用しよう。

有理数 α の無限小数表記を $0.d_1d_2d_3\cdots$（既知）とし，既約分数表記を s/t $(0 \leqq s < t < M)$（未知）とする。つまり $0 \leqq \alpha < 1$ の場合であるが，これ以外の場合も容易にこの場合に帰着できることは明らかだろう。ここでは分母 t の上限 M が分かっている場合に，この問題を $O(\mathrm{len}(M))$ で解くアルゴリズムについて述べる。まず $10^e > 2M^2$ なる e を求め，$n = 10^e$ とする。10 進表記 $d_1d_2\cdots d_e$ が表す数を b とし，$\mathrm{EEA}(n, b) = \{(r_i, s_i, t_i)\}$ を計算する。$r_j \leqq M$ となる最小の j を求めれば，$s/t = -s_j/t_j$ となる。具体的なアルゴリズムの形に書くと……

有理数の無限 (循環) 小数から既約分数表記への変換

入力 有理数 α $(0 \leqq \alpha < 1)$ の 10 進無限小数表記 $0.d_1d_2d_3\cdots$
出力 α の既約分数表記 s/t $(0 \leqq s < t)$（ただし，$t \leqq M$ なる M は既知とする）

(1) $i \longleftarrow 1$; $n \longleftarrow 1$; $b \longleftarrow 0$;
(2) **while** $n \leqq 2M^2$ **do**
　　$n \longleftarrow 10n$; $b \longleftarrow 10b + d_i$; $i \longleftarrow i + 1$;
(3) $\mathrm{EEA}(n, b) = \{(r_i, s_i, t_i)\}$ を計算し，$r_j < M$ を満たす最小の j を求める;
(4) $s \longleftarrow s_j$; $t \longleftarrow -t_j$;

定理 4.83. 上記アルゴリズムの出力 s，t により，入力 α の既約分数表記 s/t が得られる。

証明: 有理数 α の既約分数表記を s/t，無限小数表記を $0.d_1d_2d_3\cdots$ としよう。ステップ (1) と (2) により，$2M^2 < 10^e = n$ を満たす n と 10 進表記 $d_1d_2\cdots d_e$ が表す数 b が得られ，$0 \leqq b < t$ となることは，明らかだろう。また，$ns/t = 10^e\alpha$ の少数表示は $d_1d_2\cdots d_e.d_{e+1}\cdots$ だから，$b \leqq ns/t \leqq b + 1$ であり，従って $0 \leqq ns - bt \leqq t$ だから，$r = ns - bt$，$R = T = M$ とおけば，$(r, s, -t)$ は定理 4.82 の (r, s, t) に関する条件 (*) を満たす。n は $2RT = 2M^2 < n$ を満たすから，同定理の (b) より 0 でない整数 q が存在して，$r = r_j q$，$s = s_j q$，$-t = t_j q$ だが，s/t は既約分数だったから $q = 1$，従って $s = s_j$，$t = -t_j$ である。 □

例 4.84. 先の $r = 2/7 = 0.\dot{2}85714\dot{}$ を例にとる。$M = 7$ とすると，$10^2 = 100 > 98 = 2 \cdot 7^2$ だから，ステップ (2) まで進んだ段階で $n = 100, b = 28$ となる。従って $\mathrm{EEA}(100, 28)$ を実行すると，

i	q_i	r_i	s_i	t_i
0		100	1	0
1	3	28	0	1
2	1	16	1	-3
3	1	12	-1	4
4	1	4	2	-7
5	3	0	-7	25

となる。$r_j \leqq M = 7$ となる最小の j は 4 だから，$s = s_4 = 2$，$t = -t_4 = 7$ である。

例 4.85. もう少し大きな数値を扱ってみる。$r = 511/710 = 0.7197183098591549$，$M = 1000$ とする。$10^7 \geqq 2 \cdot 1000^2$ だから，ステップ (2) まで進んだ段階で $n = 10^7 = 10000000, b = 7197183$ となる。従って EEA$(10000000, 7197183)$ を実行すると，

i	q_i	r_i	s_i	t_i
0		10000000	1	0
1	1	7197183	0	1
2	1	2802817	1	-1
3	2	1591549	-2	3
4	1	1211268	3	-4
5	1	380281	-5	7
6	3	70425	18	-25
7	5	28156	-95	132
8	2	14133	208	-289
7	1	14043	-303	421
9	1	70	511	-710
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

となる。EEA の計算はこの後も続けられるが，$j = 9$ の段階で $r_j = 70 \leqq M = 1000$ となったから，ここで終了して $s = s_9 = 511$，$t = -t_9 = 710$ とすればよい。

問題 4.86. 次は有理数の 10 進小数表記を途中で打ち切ったものである。既約分数表記したとき，いずれも分母は 700 以下と分っている。元の有理数の既約分数表記 (10 進) を求めよ。

(a) 0.24610591
(b) 1.47863247
(c) 3.14159292

問題 4.87. 先のアルゴリズムは，b の代わりに $b+1$ を用いて，EEA$(n, b+1)$ を計算しても，ほぼ同様に機能する。わずかにアルゴリズムを修正する必要があるが，どのようにすれば良いか？

少し考えればわかるように，先のアルゴリズムは 10 進小数表記が与えられたときばかりでなく，わずかな変更で任意の d 進小数表記に対しても有効である。

問題 4.88. 次は有理数の 2 進小数表記を途中で打ち切ったものである。元の有理数の既約分数表記 (10 進) を求めよ。分母は $2^6 = 64$ 未満と分っている。

(a) 0.01111 10001 01011 111
(b) 1.10011 00000 11011 101

4.13 連分数 (continued fraction) と実数の有理数近似

R をユークリッド整域とし，K をその商体 R/R とする。$r_0/r_1 \in K$ $(r_0, r_1 \in R)$ に対し，互除法による計算列 $(q_i, r_{i+1}) \longleftarrow r_{i-1} \div r_i$ $(i = 1, 2, \ldots, l)$ を考え，さらに次のような分数の列を考える。

$$\frac{r_0}{r_1} = q_1 + \frac{r_2}{r_1} = q_1 + \frac{1}{\frac{r_1}{r_2}} = q_1 + \frac{1}{q_2 + \frac{r_3}{r_2}} = \cdots = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cfrac{1}{\ddots + \cfrac{1}{q_k + \frac{r_{k+1}}{r_k}}}}}$$

これを r_0/r_1 の (正則) **連分数** (continued fraction) への展開と呼び，$[q_1, q_2, \ldots, q_k, r_k/r_{k+1}]$ と表記する。特に $r_{i+1} = 0$ のとき $[q_1, q_2, \ldots, q_k]$ と表記する。一般に

$$[q_1] = q_1, \quad [q_1, q_2, \ldots, q_k] = q_1 + \frac{1}{[q_2, \ldots, q_k]}$$

が成り立つ。

例 4.89. $126/35$ の連分数展開は，

$$(3, 21) \longleftarrow 126 \div 35, \quad (1, 14) \longleftarrow 35 \div 21, \quad (1, 7) \longleftarrow 21 \div 14, \quad (2, 0) \longleftarrow 14 \div 7$$

により $[3, 1, 1, 2]$ である。逆に

$$[2] = 2, \quad [1, 2] = 1 + 1/2 = 3/2, \quad [1, 1, 2] = 1 + 2/3 = 5/3, \quad [3, 1, 1, 2] = 3 + 3/5 = 18/5$$

という計算で，元の有理数 $126/35 = 18/5$ が求まる。 □

互除法では計算途上で得られる整商 q_i には直接の関心が向かなかったのに対して，連分数展開では q_i そのものを得ることが目的であるという点が，大きな差であるが，既に互助法に関して述べた性質の多くは，連分数展開に関連した言い換えが可能である。

問題 4.90. 連分数 $[q_1, q_2, \ldots, q_k]$ に対して，行列 $Q_i \stackrel{\text{def}}{=} \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix}$ $(i = 1, 2, \ldots, k-1)$ を考える。

$$Q_1 Q_2 \cdots Q_k = \begin{pmatrix} s' & s \\ t' & t \end{pmatrix}$$

とすると $[q_1, q_2, \ldots, q_k] = -t/s$，$[q_1, q_2, \ldots, q_{k-1}] = -t'/s'$ である。

注 例 4.89 では連分数 $[q_1, q_2, \ldots, q_k]$ を通常の分数の形に戻すのに，$[q_k]$ から始めて順に左へ伸ばしながら，最終的に $[q_1, q_2, \ldots, q_k]$ の値を求めていたが，問題 4.90 によれば，$[q_1]$ から順に右に伸ばしながら $[q_1, q_2, \ldots, q_k]$ の値を求めることができる。すなわち $Q_1, Q_1Q_2, \ldots$ を順次計算し，最終的に $Q_1 Q_2 \cdots Q_k$ を求めれば，その第 2 列から直ちに $[q_1, q_2, \ldots, q_k]$ の値が求まる。

問題 4.91. 連分数 $[q_1, q_2, \ldots, q_k]$ に対して，$s_0 = 1, t_0 = 0, s_1 = q_1, t_1 = 1$ とおき

$$s_i = s_{i-1} q_i + s_{i-2}, \quad t_i = t_{i-1} q_i + t_{i-2} \ (i = 2, \ldots, k)$$

とすれば，$[q_1, q_2, \ldots, q_i] = s_i/t_i$ $(i = 1, \ldots, k)$ である。

ユークリッド整域 R の商体の場合は，全ての要素が R の元による有限の連分数展開を持つが，そうでなくとも整除とその剰余という概念が定義できるような体においては，その要素を (無限あるいは有限の) 連分数で表現できることが多い。例えば，実数体 $\mathbb{R}$ を考えよう。$\alpha = \alpha_1 \in \mathbb{R}$ とし，$q_i = \lfloor \alpha_i \rfloor$，$\alpha_{i+1} = \dfrac{1}{\alpha_i - q_i}$ という計算列を考える。α が有理数であれば，$\alpha_k = q_k$ なる k が存在するので，この計算は終了し $\alpha_1 = [q_1, q_2, \ldots, q_k]$ と表現されるが，α が無理数であれば，計算はいつまでも続き，$\alpha = [q_1, q_2, \ldots]$ という無限連分数展開が得られる。

例 4.92. $\alpha = \sqrt{3}$ の場合，

$$q_1 = \left\lfloor \sqrt{3} \right\rfloor = 1, \quad \alpha_2 = \frac{1}{\sqrt{3}-1} = \frac{\sqrt{3}+1}{2},$$
$$q_2 = \left\lfloor \frac{\sqrt{3}+1}{2} \right\rfloor = 1, \quad \alpha_3 = \frac{1}{\frac{\sqrt{3}+1}{2} - 1} = \sqrt{3}+1,$$
$$q_3 = \left\lfloor \sqrt{3}+1 \right\rfloor = 2, \quad \alpha_3 = \frac{1}{\sqrt{3}-1} = \frac{\sqrt{3}+1}{2}, \quad \ldots$$

以下同じ計算が繰り返されるので，$\sqrt{3}$ の連分数展開は循環する無限列 $[1, 1, 2, 1, 2, 1, 2, \ldots]$ である。この循環節を上線で示し，$[1, \overline{1, 2}]$ と表記する。

実数を連分数展開して循環する無限列となるための必要十分条件は，その実数が $\dfrac{m \pm \sqrt{n}}{q}$ $(m, n, q \in \mathbb{Z})$ という形に書けることである[5]。

実数を有理数で近似すると問題を一般に**有理近似**とか**ディオファンタス近似**という。無理数の連分数展開は，有限にならないが，それを途中で切ってできた有限連分数は，当然有理数であり，もとの実数の優れた近似を与えるという点で有用である。

問題 4.93. $n, b \in \mathbb{Z}$ は，$0 \leqq b < n$ を満たすとし，$\mathrm{EEA}(n, b) = \{(r_i, s_i, t_i)\}_{i=0}^{\lambda+1}$ とする。$\varepsilon_i = b/n + s_i/t_i$ とおくとき，次を示せ。

(a) $\varepsilon_i = \dfrac{r_i}{n t_i}$
(b) ε_i の絶対値は減少列をなし，符号は交互に変わる。
(c) $i = 1, \ldots, \lambda$ に対して $|\varepsilon_i| < \dfrac{1}{{t_i}^2}$ である。
(d) $t \neq 0$ で $|\dfrac{b}{n} - \dfrac{s}{t}| < \dfrac{1}{2t^2}$ ならば，ある i について $\dfrac{s}{t} = -\dfrac{s_i}{t_i}$ である。(ヒント: 定理 4.82 を用いよ)
(e) $0 < |t| \leqq |t_i|$ で $|\dfrac{b}{n} - \dfrac{s}{t}| < |\varepsilon_i|$ ならば，$\dfrac{s}{t} = -\dfrac{s_i}{t_i}$ である。(ヒント: 定理 4.82 を用いよ)

例 4.94. 円周率 $\pi = 3.14159265358979\ldots$ の連分数展開は $[3, 7, 15, 1, 292, \ldots]$ であるが，

$$[3] = 3, \quad [3, 7] = 22/7, \quad [3, 7, 15] = 333/106, \quad [3, 7, 15, 1] = 355/113$$

は，いずれもその近似値として昔から使われてきた有理数である。

[5] このような実数を 2 次無理数といい，整数係数の 2 次方程式の解となる

問題 4.95.
次の数を連分数に展開せよ。有理数は有限連分数，無理数は循環連分数として表記せよ。

(a) $\dfrac{79}{561}$
(b) $2+\sqrt{7}$

問題 4.96.
循環連分数 $r=[2,\overline{1}]$ を通常の 2 次無理数として表記せよ。また，分母が 1 以上 30 以下で r にもっとも近い有理数を求めよ。

問題 4.97.
n を正の整数とするとき， 循環連分数 $[\overline{n}]=[n,n,n,\ldots]$ を通常の 2 次無理数として表記せよ。また，循環連分数 $[\overline{n,1}]=[n,1,n,1,\ldots]$ ならどうか。

問題 4.98.
m を正の整数とするとき，$\sqrt{1+m^2}$ を（循環）連分数表記するとどうなるか？

問題 4.99. 実数 α の連分数展開は，偶数番目の項がすべて偶数となり，$[q_1,2q_2,q_3,2q_4,\ldots]$ のように書けるとする。このとき実数 2α の連分数展開はどうなるか？

問題 4.100. n 正の整数とするとき，循環連分数 $[\overline{n}]$ を通常の 2 次無理数の形に書け。

4.13.1 カレンダー (Calendars) ★

連分数による実数の有理近似の応用例として，カレンダーについて触れよう。

春分とは，春に太陽が天の赤道を通過する瞬間をいう[6]。太陽年 (tropical year, solar year) とは，ある春分から次の春分までの期間をいい，現在ほぼ世界中で用いられている暦 (グレゴリオ暦, the Gregorian calendar) と分かちがたく結びついている。1 太陽年の長さは，だいたい 365.242190 日，すなわち 365 日 5 時間 48 分 45.2 秒 である。(正確には，その値は 1 世紀ごとに約 0.53 秒ずつ小さくなっているが，ここでそのことを考慮する必要はないだろう。)

文明が芽生えて以来，人々は，地球を廻る太陰 (天体の月, moon) の運行や季節の周期性を表現するためにカレンダーを用いてきた。太陰暦 (lunar calender) は，時間を月 (month) に分割するが，それは元々は新月を毎月の始まりにするというものであった。1 太陰月は 29 日と 30 日の間なので，太陰暦は季節とはうまく同期しない。一方，太陽暦は，太陰の満ち欠けを無視し，季節をなるべく正確に会わせようとするものである。

初期のローマ帝国で使われていたカレンダーは，太陰太陽暦と呼ぶべきもので，1 年は，元々は 10 ヶ月，後には 12 ヶ月からなり，ときおり閏月として，余分な 1 月を追加して季節をあわせるというものだった[7]。ローマ帝国では，ある時期から，太陽暦を採用したが 1 年を 365 日と決めカレンダーの閏年調整を全くしなかったので，季節がカレンダーと大きく狂って来た。紀元前 46 年以降，時の皇帝だったユリウス・カエサル (ジュリアス・シーザー，Julius Caesar) の名前を冠したユリウス暦 (the Julian

[6] その瞬間が含まれる日を「春分の日」といい，日本ではおおむね 3 月 21 日である
[7] 日本の江戸時代まで使われていた旧暦も，(中国のカレンダーをまねたもので) 閏月のはさみ方に関する細かい規則に特徴はあるが，同様の太陰太陽暦である

calendar) が使われるようになったが，それに併せて，季節の調整を一気に行なったために，紀元前 46 年は 445 日もあり,「錯乱の年 (ラテン語で annus cofusionis)」と呼ばれるほどである。

ユリウス暦では，1 年を約 365.25 日として，4 年に 1 度，閏日を追加する。これは，真の太陽年の長さに相当に近いものであるが，なお季節とのずれが 400 年で 3 日ほど生ずる。

16 世紀の終わりごろになると，ユリウス暦でも季節とのずれが大きくなった。春分は，3 月 10 日になり，本当の日付であるべき 3 月 21 日と大きく隔たってしまった。これを正すために，1582 年 2 月 24 日に時のローマ法王グレゴリー 13 世 (Pope Gregory XIII) が改暦の教書を発し，まず，同年 10 月 4 日の翌日を 10 月 15 日にすることで，そのときまでの誤差の蓄積を修正した。次に，閏年の規則を変更して，100 で割り切れるが 400 で割り切れない年は平年とすることにした。この改暦は，カトリック以外の国ではすぐには実行されなかったが，次第に広まり，現在も世界中で使われているグレゴリオ暦となった。グレゴリオ暦では，例えば，1700 年，1800 年，1900 年は平年であるが，2000 年は閏年になる。この変更により，ユリウス暦では 400 年で 100 回あった閏年が 97 回に減り，カレンダーはより正確に季節と合うようになった。1 太陽年は約

$$365\frac{97}{400} = 365.2425$$

日とされている。これは真の太陽年より約 26.8 秒長く，3200 年くらいで 1 日狂う。

そこで，真の太陽年の有理近似を連分数を利用して求めてみよう。

$$365.242190 = [365, 4, 7, 1, 3, 24, 6, 2, 2]$$

である。連分数の途中までをとった近似値と誤差を表にすると，次のようになる。

連分数	分数	小数	年間誤差
$[365, 4]$	$365 + 1/4$	365.25	11 分 14.8 秒
$[365, 4, 7]$	$365 + 7/29$	$365.24137\cdots$	-1 分 10.0 秒
$[365, 4, 7, 1]$	$365 + 8/33$	$365.24242\cdots$	20.2 秒
$[365, 4, 7, 1, 3]$	$365 + 31/128$	$365.24218\cdots$	-0.2 秒

表の最初の近似値はユリウス暦で使われたものである。グレゴリオ暦の近似値 $365\frac{97}{400}$ はこの表には現れないが，3 番目の $365\frac{8}{33}$ でも，既にグレゴリオ暦より良い近似値を与えている。最後の近似値は，分母の 128 が 2^7 なので，計算機科学者には大変魅力的である。しかも,「4 で割り切れて 128 で割り切れない年は閏年」という簡単な規則で実現できて，なんと誤差は年に約 0.2 秒であるから，1 日の狂いが生ずるには 40 万年ほどもかかり，半永久的に使える規則になる。

4.13.2 音階 (Musical scales) ★

連分数のもう一つの応用が音楽理論にある。音程とは，(例えば，G–C のように異なる高さの) 2 つの音の間隔をいう。この用語は，また，ある楽器でその 2 つの音を次々にまたは同時に引いたときに発生する調べを意味する。各音程に対応するものとして，その 2 つの音の周波数の比がある。よく知られた

音程の表を次に挙げる。

周波数の比	名前	例
$r_1 = 2:1$	オクターブ (8 度)	c–C
$r_2 = 3:2$	5 度	G–C
$r_3 = 4:3$	4 度	F–C
$r_4 = 5:4$	長 3 度	E–C
$r_5 = 6:5$	短 3 度	E♭–C
$r_6 = 9:8$	全音 (長 2 度)	D–C

音程による「計算」は次のように行なえばいい。2 つの音程の組み合わせに対応するのは，周波数比の積である。例えば，オクターブ c–C は，5 度 c–F と 4 度 F–C の組み合わせと考えることができ，実際 $2/1 = (3/2)\cdot(4/3)$ である。

音楽理論の起源は初期のピタゴラス派の時代にまでさかのぼる。彼らは，固定長の弦 1 本だけからなる弦楽器モノコードで実験していた。モノコードの弦は，移動可能な柱によって，2 つの可変長の部分に分けられる。その実験により，例えば，1/2, 2/3, 3/4 の長さの弦を長さ 1 の弦とともに奏でると，人間の耳には魅力的な音程となることが発見し，もっと一般的には，弦の長さの比 (すなわち音の周波数の比) が小さな正の整数の比であるときに，そうなるという主張を提示した。

ピタゴラス派の調律理論は，全ての音程の周波数比を，5 度に対する比 3/2 とオクターブに対する比 2/1 から導く。ディディモスが発明した全音階調律は，加えて長 3 度の周波数比 5/4 の使用を定着させた。ハ長調の音階 C–D–E–F–G–A–B–c ついて，基音 C との周波数比を表にすると次のようになる。

音	C	D	E	F	G	A	B	c
ピタゴラス調律	1:1	9:8	81:64	4:3	3:2	27:16	243:128	2:1
全音階調律	1:1	9:8	5:4	4:3	3:2	5:3	15:8	2:1

ピタゴラス調律では，周波数比で 8/9 の全音 (長 2 度) の差が D–C，E–D, G–F, A–G, B–A の 5 箇所にあり，周波数比で 256/243 の半音の差が F–E と c–B の 2 箇所にある。周波数比の分子と分母に現れる全ての数が 2 と 3 の冪である。全音階調律では，比が 9/8 の (大) 全音差が D–C，G–F, B–A の 3 箇所，比が 10/9 の (小) 全音差が E–D, A–G の 2 箇所，比が 16/15 の半音差が F–E と c–B の 2 箇所である。

ピタゴラス調律でも全音階調律でも，一つ具合の悪いことは，例えばハ長調からニ長調へというように，一つの調で書かれた楽曲を別の調に移調する場合である。それは (全体の周波数比は変わらないように) 全ての音の周波数に音程 D–C に当たる比を掛けることに相当するが，ハ長調に調律されたピアノでは，人間の声ほどには簡単にいかない。例えば，全音階調律の場合，ハ長調の全音差 E–D の周波数比は $(5/4)/(9/8) = 10/9$ で，二長調でそうあるべき 9/8 ではない。この問題を解決しようと，多くの数学者や音楽家が工夫を凝らして来たが，近年，支配的な調律法は 12 平均律である。

この音階法はオクターブの音程 c–C を，等しく 12 の半音差に分けるもので，例えば，ピアノでは C から c までの隣り合った鍵盤がそれぞれ半音の差からなる音階を形成する。8 つの白鍵がハ長調の音階 C–D–E–F–G–A–B–c に対応し，黒鍵をはさむ 5 箇所が全音の差，黒鍵をはさまない 2 箇所が半音の差となっている。12 という数が選ばれたのはどうしてだろうか？

オクターブを均等に n 個の半音に分けると，半音 1 つの周波数比は $2^{1/n}$ となるが，そのような半音の整数倍で心地よいとされる音程を作りたいとする。それは，$r_1 = 2/1$，$r_2 = 3/2$，$r_3 = 4/3$，$r_4 = 5/4$，

$r_5 = 6/5$，$r_6 = 9/8$ が，ある整数 d_i によって近似的に $r_i = 2^{d_i/n}$ と表せるということを意味する。しかしながら，例えば $r_2 = 3/2$ の場合，$r_2 = 2^{d_2/n}$ となる整数 d_2, n は存在しないから，対数をとり，

$$\left| \lg r_i - \frac{d_i}{n} \right|$$

をなるべく小さくするような d_i を見つけることが課題となる。これは，有理近似の問題であり，$\lg r_i$ の連分数展開よって解くのが一番いい。

例えば $\lg r_5 = \lg(5/6) = 0.2630344058\ldots$ を連分数展開すると $[0, 3, 1, 4, 22, 4, \ldots]$ となり，

$$\begin{aligned}
[0, 3] &= 1/3 = 0.33333\ldots \\
[0, 3, 1] &= 1/4 = 0.25000\ldots \\
[0, 3, 1, 4] &= 5/19 = 0.26315\ldots \\
[0, 3, 1, 4, 22] &= 111/422 = 0.26303\ldots
\end{aligned}$$

がその近似分数として得られる。平均律の問題は，r_5 のみでなく，いくつかの i に対して $\left| \lg r_i - \frac{d_i}{n} \right|$ の値を小さくできる共通の n を見つけなければならないことだ。同時有理近似についての詳細は省略するが，

$$\begin{aligned}
\lg r_1 &= \lg(2/1) = 1.00000\ldots \approx [1] = 12/12 \\
\lg r_2 &= \lg(3/2) = 0.58496\ldots \approx [0, 1, 1, 2, 2] = 7/12 \\
\lg r_3 &= \lg(4/3) = 0.41503\ldots \approx [0, 2, 2, 2] = 5/12 \\
\lg r_4 &= \lg(5/4) = 0.32192\ldots \approx [0, 3] = 4/12 \\
\lg r_5 &= \lg(6/5) = 0.25303\ldots \approx [0, 3, 1] = 3/12 \\
\lg r_6 &= \lg(9/8) = 0.16992\ldots \approx [0, 5, 1] = 2/12
\end{aligned}$$

となり，共通分母 $n = 12$ は，あまり悪くない近似を与える。

5 反覆法

f を $\mathbb{R}$（または $\mathbb{C}$）上の関数とし，方程式 $f(x)=0$ を考える。f が多項式でないとき，または多項式であっても次数が 5 以上のとき，一般の場合を代数的に解く手段はない。しかし，実用的には十分な精度で解が求まれば良いことも多い。そのための手段は，通常，**反復法**である。つまり，適当な数値 x_0 から出発して，解 α に収束する列 $x_0, x_1, \ldots$ をつくり，x_n が α に十分近づいた段階で計算を打ち切り，x_n を近似解とする。計算を打ち切る条件は，色々考えられるが，本講義ノートでは問題とせず，主に反覆列を作り出す方法について議論する。

5.1 縮小写像の原理

まず，一般の距離空間 X 上で定義された関数 $g : X \to X$ の不動点 α に，x_0 から始めた反覆 $x_{n+1} = g(x_n)$ が収束するための条件を論じよう。

定理 5.1 (縮小写像の原理)**.**
(X, d) を完備距離空間とし，I をその閉部分集合とする。関数 $g : X \to X$ が次の条件 (a) を満たし，さらに (b) を満たす $\lambda \in (0..1) \subset \mathbb{R}$ が存在すれば，g の不動点は，I 内にちょうど 1 つ存在する。それを α とすると，$x_0 \in I$，$x_{n+1} = g(x_n)$ とするとき，α は極限 $\lim_{i\to\infty} x_i$ として得られる。

(a) $g(I) \subset I$
(b) $\forall x, y \in I\ d(g(x), g(y)) \leqq \lambda d(x, y)$

上の条件を満たす写像 g を**縮小写像**と呼ぶ。条件 (b) を **Lipschitz 条件**と呼び，(b) を満たす λ を **Lipschitz 定数**という。さらに反復の収束の速さについて，$d(x_n, \alpha) \leqq \lambda^n d(x_0, \alpha)$ が成り立つ。
証明: 点列 $x_0, x_1, \ldots$ を上のように定義すると

$$d(x_{n+1}, x_n) = d(g(x_n), g(x_{n-1})) \leqq \lambda d(x_n, x_{n-1}) \leqq \cdots \leqq \lambda^n d(x_1, x_0)$$

ゆえに，$m > n$ なら

$$d(x_m, x_n) \leqq d(x_m, x_{m-1}) + \cdots + d(x_{n+1}, x_n) \leqq (\lambda^{m-1} + \cdots + \lambda^n) d(x_1, x_0) = \frac{\lambda^n - \lambda^m}{1-\lambda} d(x_1, x_0)$$

である。よって，$d(x_m, x_n) \to 0\ (m, n \to \infty)$ だから $x_0, x_1, \ldots$ は I 内のコーシー列であり，I は完備距離空間の閉部分集合だから，I 内のある点 α に収束する。α は g の不動点である。実際，(b) より g は連続だから

$$g(\alpha) = g(\lim_{n\to\infty} x_n) = \lim_{n\to\infty} g(x_n) = \lim_{n\to\infty} x_{n+1} = \alpha$$

である。また，β を I 内の別の不動点とすれば

$$0 < d(\alpha, \beta) = d(g(\alpha), g(\beta)) \leqq \lambda d(\alpha, \beta) < d(\alpha, \beta)$$

により，矛盾する。 □

問題 5.2. 定理 5.1 の写像 g は I で一様連続（よって当然連続）であることを示せ。

問題 5.3. 定理 5.1 で $\varepsilon_n = d(x_n, x_{n+1})$ とすれば

$$\frac{\varepsilon_n}{1+\lambda} \leqq d(x_n, \alpha) \leqq \frac{\varepsilon_n}{1-\lambda} \leqq \frac{\lambda^n \varepsilon_0}{1-\lambda} \quad (n \geqq 1)$$

が成立することを証明せよ。

系 5.4.
α が実数関数 $g : \mathbb{R} \to \mathbb{R}$ の不動点で，$g(x)$ が閉区間 $I = [\alpha - d..\alpha + d]$ 内において C^1 級とする。$\max_{x \in I}\{|g'(x)|\} \leqq \lambda$ を満たす λ $(0 \leqq \lambda < 1)$ が存在するならば，α は，I 内のただ 1 つの g の不動点であり，$x_0 \in I$，$x_{n+1} = g(x_n)$ とするとき，極限 $\lim_{i \to \infty} x_i$ として得られる。
証明: I は完備距離空間 $\mathbb{R}$ の閉部分集合だから，定理 5.1 の条件を確認すればよい。条件 (a) を示す。$x \in I$ とすれば，平均値の定理より，ある $\xi \in I$ について

$$g(x) - \alpha = g(x) - g(\alpha) = g'(\xi)(x - \alpha)$$

だから，$|g(x) - \alpha| \leqq |g'(\xi)||x - \alpha| \leqq \lambda |x - \alpha| \leqq |x - \alpha| \leqq d$ により $g(x) \in I$ となる。条件 (b) も同様で，任意の $x, y \in I$ について，平均値の定理より，

$$|g(x) - g(y)| = |g'(\xi)||x - y| \leqq \lambda |x - y|$$

である。 □

5.2 Newton 法

今，f を $\mathbb{R}$ または $\mathbb{C}$ 上で定義された関数とし，$f(x) = 0$ の解を範囲 I 内で探すとしよう。すると，その解を求める一般的な戦略として，$f(x) = 0$ と $g(x) = x$ とが I 内で共通の解を持つような関数 g を見つけ，g の不動点（の近似値）を反復法で計算するということが考えられる。さらに，適当な関数 $\varphi(x)$ を考え $g(x) \stackrel{\text{def}}{=} x - \varphi(x) f(x)$ とおくと，$f(x) = 0$ ならば x は $g(x)$ の不動点である。逆に $g(x) = x$ ならば，$\varphi(x) f(x) = 0$ だから，$\varphi(x) = 0$ でなければ $f(x) = 0$ だ。つまり，I の範囲では 0 にならない関数 φ を見つけることで，g の候補 $x - \varphi(x) f(x)$ が得られる。

φ としてはいろいろなものが考えられるので例をあげよう。

(1) **von Mises 法** $\varphi(x) = \dfrac{1}{f'(x_0)}$
(2) **線型逆補間法** $\varphi(x) = \dfrac{x - c}{f(x) - f(c)}$
(3) **Newton 法** $\varphi(x) = \dfrac{1}{f'(x)}$

線型逆補間法は f が連続というだけで使えるという点で有望な方法であるが，ここでは収束速度で圧倒的に優れた Newton 法を詳しく見ることにする。Newton 法は f が微分可能であることを要求するが，例外的な場合を除き，2 次収束する非常に優れた方法である。$g(x) = x - \dfrac{f(x)}{f'(x)}$ を反復関数として用いるが，これは幾何学的には，点 $(x_n, f(x_n))$ における f の接線と x 軸との交点の x 座標を次の x_{n+1} とする方法である。

定義 5.5. 一般に $k < 1$ が存在して，α に収束する列 $x_0, x_1, \ldots$ が十分大きな n に対して

$$|x_{n+1} - \alpha| \leqq k|x_n - \alpha|$$

を満たすとき，**線形収束**する，または **1 次収束**するという。例えば，縮小写像の原理（定理 5.1） による収束は 1 次収束である。さらに

$$\lim_{n \to \infty} \frac{x_{n+1} - \alpha}{x_n - \alpha} = 0$$

のとき**超 1 次収束**するという。また，実数 $M \in \mathbb{R}$ が存在して十分大きな n に対して

$$|x_{n+1} - \alpha| \leqq M|x_n - \alpha|^p$$

を満たすとき（すなわち $x_{n+1} - \alpha = O((x_n - \alpha)^p)$ のとき），p **次収束**するという。

定理 5.6. 関数 f は，$f(x) = 0$ の解 α を含む適当な区間 I で C^2 級であり，かつ $f'(x) \neq 0$ とする。このとき，反覆式 $g(x) = x - \dfrac{f(x)}{f'(x)}$ により α に十分近い点 x_0 から始める Newton 反復 $x_{n+1} = g(x_n)$ は α に 2 次収束する。

証明:

$$g'(x) = 1 - \frac{f'(x)^2 - f(x)f''(x)}{f'(x)^2} = \frac{f(x)f''(x)}{f'(x)^2}$$

である。よって $g'(x)$ は連続であり $g'(\alpha) = 0$ だから，α の近辺では $|g'(x)|$ は十分小さくなる。従って，$[\alpha - d..\alpha + d] \subset I$ なる適当な $d > 0$ をとれば，g は系 5.4 の仮定を満たすから，反覆 $x_{n+1} = g(x_n)$ は α に収束する。

さらに f を $x = x_n \in I$ の周りで 2 次までテーラー展開して，$x = \alpha$ を代入すると

$$0 = f(\alpha) = f(x_n) + f'(x_n)(\alpha - x_n) + \frac{1}{2}f''(\xi)(\alpha - x_n)^2 \quad (\xi \in I)$$

だから，反復式によって

$$x_{n+1} - \alpha = g(x_n) - \alpha = x_n - \alpha - \frac{f(x_n)}{f'(x_n)} = \frac{f''(\xi)}{2f'(x_n)}(x_n - \alpha)^2$$

である。区間 I において，$f'(x)$ は連続で $f'(x) \neq 0$ だから $|f'(x)|$ の最小値を A とし，$f''(x)$ は連続だから $|f''(x)|$ の最大値を B とすれば，

$$|x_{n+1} - \alpha| \leqq \frac{B}{2A}|x_n - \alpha|^2$$

が成り立つから，これは 2 次収束である。 □

5.3 加速法

数列 $\{x_n\}$ が α に収束するとき，$\{x_n\}$ から計算される別の数列 $\{y_n\}$ があって，

$$\lim_{n \to \infty} \frac{y_n - \alpha}{x_n - \alpha} = 0$$

（すなわち $y_n - \alpha = o(x_n - \alpha)$）ならば，$y_n$ を計算することで収束は加速される。

$k_n \stackrel{\text{def}}{=} \dfrac{x_{n+1}-\alpha}{x_n-\alpha}$ とするとき，$\lim_{n\to\infty} k_n = k < 1$ であれば，明らかに $\{x_n\}$ は 1 次収束する。このとき，y_n を

$$y_n = x_n - \frac{(x_{n+1}-x_n)^2}{x_{n+2}-2x_{n+1}+x_n} \left(= x_{n+2} - \frac{(x_{n+2}-x_{n+1})^2}{x_{n+2}-2x_{n+1}+x_n} \right)$$

とする方法を **Aitken 加速法**と呼ぶ。

定理 5.7. $k > 0$ なら Aitken 加速 $\{y_n\}$ は，$y_n - \alpha = o(x_{n+2}-\alpha)$ の意味で $\{x_{n+2}\}$ を加速する。
証明: $e_n = x_n - \alpha$ とおく。y_n の定義より

$$y_n - \alpha = e_n - \frac{(e_{n+1}-e_n)^2}{e_{n+2}-2e_{n+1}+e_n} = \frac{e_n e_{n+2} - {e_{n+1}}^2}{e_{n+2}-2e_{n+1}+e_n}$$

$$\frac{y_n-\alpha}{e_{n+2}} = \frac{k_{n+1}k_n - {k_n}^2}{k_{n+1}k_n(k_{n+1}k_n - 2k_n + 1)} = \frac{1-k_n/k_{n+1}}{k_{n+2}k_n - 2k_n + 1}$$

だから，$\lim k_n = k > 0$ より $y_n - \alpha = o(e_{n+2})$ である。 □

問題 5.8. $k = 0$，すなわち $\{x_n\}$ が超 1 次収束するならば，Aitken 加速 $\{x_n\}$ は $\{x_{n+2}\}$ を加速するとは限らない。そのような例 $\{x_n\}$ を与えよ。

問題 5.9. $k = 0$ であっても Aitken 加速 $\{y_n\}$ は $\{x_n\}$ を加速することを示せ。（ただし，y_n を求めるには x_{n+2} までの計算が必要だから，これは $k = 0$ の場合に Aitken 加速が有効であるという根拠にはならない。）

問題 5.10. $0 < |r| < 1$ とする。等比級数 $x_n = \sum_{i=0}^{n} r^i$ が 1 次収束することを示せ。$\{x_n\}$ を Aitken 加速するとどうなるか？

5.4 Steffensen 反復

Steffensen は実数関数 $g : \mathbb{R} \to \mathbb{R}$ の不動点を求めるのに，反復

$$z_{n+1} = \frac{z_n g(g(z_n)) - g(z_n)^2}{g(g(z_n)) - 2g(z_n) + z_n} = g(g(z_n)) - \frac{[g(g(z_n)) - g(z_n)]^2}{g(g(z_n)) - 2g(z_n) + z_n}$$

を提案した。これは形の上では $x_n = z_n$，$x_{n+1} = g(x_n)$，$x_{n+2} = g(x_{n+1})$ に Aitken 加速したものに等しいが，その計算結果そのものを反復に用いている。これが優れているのは，反復列 $x_{n+1} = g(x_n)$ が発散する場合でも，初期値 z_0 を十分近くに選べば $\{z_n\}$ は 1 次以上の収束をすることだ。

定理 5.11. $g(x)$ が C^1 級で α をその不動点とする。$g'(\alpha) \neq 1$ ならば，α に十分近い z_0 から Steffensen 反復を行うと $\{z_i\}$ は α に超 1 次収束する。特に $g(x)$ が C^2 級なら 2 次収束する。
証明: 計算を簡単にするために $h(x) = g(x+\alpha) - \alpha$ とおくと，h の不動点はずれて 0 になる。また，反復式を

$$G(z) = \frac{zg(g(z)) - g(z)^2}{g(g(z)) - 2g(z) + z}, \quad H(z) = \frac{zh(h(z)) - h(z)^2}{h(h(z)) - 2h(z) + z}$$

とおくと $h(h(x)) = g(h(x)+\alpha)-\alpha = g(g(x+\alpha))-\alpha$ より $H(z) = G(z+\alpha)-\alpha$ である。$h'(0) = g'(\alpha) = c$ とおけば

$$\begin{aligned} h(z) &= cz + o(z) \\ h(h(z)) &= c(cz + o(z)) + o(cz + o(z)) = c^2 z + o(z) \\ h(h(z)) - 2h(z) + z &= c^2 z + o(z) - 2cz - o(z) + z = (c-1)^2 z + o(z) \\ h(z)^2 &= c^2 z^2 + o(z^2) \\ zh(h(z)) - h(z)^2 &= o(z^2) \end{aligned}$$

だから, $c \neq 1$ より，$H(z) = o(z)$ である。よって，Steffensen の反復式より

$$z_{n+1} - \alpha = G(z_n) - \alpha = H(z_n - \alpha) = o(z_n - \alpha)$$

となり，$\{z_i\}$ は α に超 1 次収束する。特に g が C^2 級ならば，$h(x) = cx + O(x^2)$ と書けるから，同様に $H(z) = O(z^2)$，$z_{n+1} - \alpha = O((z_n - \alpha)^2)$ を得る。 □

問題 5.12. 定理 5.11 で g が C^2 級の場合には，$\{z_i\}$ が 2 次収束することを厳密に示せ。

6 FFT と高速乗算法

本節では，多項式同士の乗算を高速に行なう方法について，考察する。2.2 節で多項式の表現とその計算を素朴に行なう方法について議論したが，n 次の多項式同士の加算には n に比例する計算量を，また乗算には n^2 に比例する計算量を要した。しかし，高速フーリエ変換（the fast Fourier transformation, FFT）を応用することで，$n \lg n$ に比例する計算量で，多項式同士の乗算が可能になる。

FFT は，代表的応用が信号処理（signal processing）であり，周期的な関数を解析するために発展して来た技術である。豊富な工学的応用を持つが，その種の応用は他書に譲り，ここでは高速乗算技術に集中して議論する。

本章を通じて，K で一般の体を，$\mathbb{C}$ で複素数体を表わす。

6.1 多項式の関数値表現

2.2 節で，計算機内における多項式の表現法について述べた。それは多項式の各係数を次数の低いほうから並べるという素朴な方法であり，それを**係数表現**（coefficient representation）と呼ぶ。その計算を素朴に行なう方法について議論したが，

しかし，多項式を計算機内で表現する方法は他にも，もう 1 つの代表的なものは**関数値表現**（point-value representation）である。これは，n 次以下の多項式 $a(x)$ を $n+1$ 個の値の対

$$\langle x_0, a(x_0)\rangle, \langle x_1, a(x_1)\rangle, \ldots, \langle x_n, a(x_n)\rangle$$

で表すもので，各 x_i の値は互いに異なるものとする。値 $x_0, \ldots, x_n$ の組合わせは自由であり，いわばその表現の**基底**をなす。

多項式 $a(x)$ の係数表現から，x_0 における値 $a(x_0)$ を計算することを**評価** (evaluation) と呼ぶ。評価は，原理的には簡単であり，2.2.2 節で述べたホーナー法によれば，点と値の組 $\langle x_i, a(x_i)\rangle$ を 1 組得るのに $O(n)$ の計算量がかかるので，（ある基底に関する）$a(x)$ の関数値表現，すなわち $n+1$ 組 $\langle x_0, a(x_0)\rangle, \ldots, \langle x_n, a(x_n)\rangle$ を得るには，通常 $O(n^2)$ かかる。

逆に（ある基底に関する）関数値表現から係数表現を求めることを**内挿** (interporation，補間) と呼ぶ。行列

$$V = \begin{pmatrix} 1 & x_0 & \cdots & {x_0}^n \\ 1 & x_1 & \cdots & {x_1}^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & {x_n}^n \end{pmatrix}$$

を考えると，点 $x_0, \ldots, x_n$ での評価とは，容易に分かるようにこの V に右から係数ベクトル $(a_0 \ \cdots \ a_n)^\top$ を掛けることに他ならない。従って，関数値表現 $\langle x_0, y_0\rangle, \ldots, \langle x_n, y_n\rangle$ が与えられているとすると，内挿とは $V(a_0 \ \cdots \ a_n)^\top = (y_0 \ \cdots \ y_n)^\top$ を満足する $a_0, \ldots, a_n$ を求めることである。V は，いわゆる Vandermond 行列であり，x_i が互いに異なるなら，正則であることが知られている。従って，次の内挿の一意性定理が成り立つ。

定理 6.1 (内挿の一意性)**.**
x_i が互いに異なるなら，n+1 個のどんな対 $\{\langle x_0, y_0\rangle, \langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle\}$ に対しても，$y_0 = a(x_0), \ldots, y_n = a(x_n)$ となる n 次以下の多項式が $a(x)$ が唯一つ定まる。

上の定理より，次のような簡便な内挿式が得られる。

定理 6.2 (Lagrange の補間公式)**.**
$a(x)$ を n 次の多項式とする。$x_0, \ldots, x_n$ が互いに異なるならば，次が成り立つ。

$$a(x) = \sum_{k=0}^{n} a(x_k) \prod_{j \neq k} \frac{x - x_j}{x_k - x_j}$$

証明: $L_k(x) \stackrel{\text{def}}{=} \prod_{j \neq k} \frac{x - x_j}{x_k - x_j}$ とすると $L_k(x_j) = [k = j]$ であることが容易に分かるが，それゆえ各 x_k における補間公式の右辺の値は $a(x_k)$ に等しい。ゆえに，内挿の一意性により，補間公式の右辺は $a(x)$ に等しいことが分かる。 □

問題 6.3. Lagrange の補間公式を用いて内挿を $O(n^2)$ で行なうアルゴリズムを示せ。（ヒント：まず，$\prod_j (x - x_j)$ を計算し，必要なら $x - x_k$ で割ることを考えよ。）

上の問題により，内挿も $O(n^2)$ 時間で行なえるから，次のように整理できる。

定理 6.4.
1 変数 n 次多項式 $a(x) \in K[x]$ の評価や内挿は，K 上の演算回数にして $O(n^2)$ の計算量で可能である。

しかし，あとで見るように，基底を巧妙に選ぶことで，評価を $O(n \lg n)$ で行なうことが可能になる。また，内挿も $O(n \lg n)$ 時間で行なうことが可能である。

基底を共有する 2 つの多項式 $a(x)$ と $b(x)$ の関数値表現

$$\langle x_0, a(x_0) \rangle, \langle x_1, a(x_1) \rangle, \ldots, \langle x_n, a(x_n) \rangle, \qquad \langle x_0, b(x_0) \rangle, \langle x_1, b(x_1) \rangle, \ldots, \langle x_n, b(x_n) \rangle$$

があれば，$a(x) + b(x)$ や $a(x)b(x)$ の関数値表現を計算するのは易しい。すなわち

$$\langle x_0, a(x_0) + b(x_0) \rangle, \langle x_1, a(x_1) + b(x_1) \rangle, \ldots, \langle x_n, a(x_n) + b(x_n) \rangle,$$
$$\langle x_0, a(x_0)b(x_0) \rangle, \langle x_1, a(x_1)b(x_1) \rangle, \ldots, \langle x_n, a(x_n)b(x_n) \rangle$$

がその表現であり，$O(n)$ 時間で計算できる。ただし，乗算の場合，多項式の積が n 次以下にならないと，上の関数値表現からはその積が一意に定まらないかもしれないことを注意する必要があるが，$a(x)$，$b(x)$ が関数値表現があらかじめ $2n$ という十分大きなサイズの基底について計算してあっても，計算量はせいぜい 2 倍にしかならない。従って，後で述べるように，表現の変換にかかる時間がともに $O(n \lg n)$ であるなら，次の手順により，係数表現を持つ 2 つの多項式の乗算が全体でも $O(n \lg n)$ で可能になる。

(Step 1) $a(x)$ と $b(x)$ をともに関数値表現に変換する。(計算量 $O(n \lg n)$)
(Step 2) 関数値表現の $a(x)$ と $b(x)$ を掛け合わせる。(計算量 $O(n)$)
(Step 3) $a(x)b(x)$ の関数値表現を係数表現に変換する。(計算量 $O(n \lg n)$)

問題 6.5. 多項式の $a(x)$ の係数表現を $(a_0, a_1, \ldots, a_n)$ とするとき，その逆順 $(a_n, \ldots, a_1, a_0)$ で表現される多項式を $a^{\text{rev}}(x)$ とする。$a(x)$ の関数値表現から $a^{\text{rev}}(x)$ の関数値表現を計算する方法について論ぜよ。ただし，関数値表現の基底に 0 は含まれていないものとする。

問題 6.6. 関数値表現を持つ 2 つの多項式の商を計算するのに，対応する関数値を「素朴に」割ると，何が問題になるかについて論ぜよ。 そのやり方で多項式の商がうまく計算されるのはどういう場合か？

6.1.1 秘密共有 (Secret Sharing)

評価や内挿の応用として秘密共有という問題を考える。$n+1$ 人に秘密を分け与えたいとしよう。秘密は適当な体 K を取って，K の要素 a_0 の形に表現されているとする。このとき，a_0 に加えて，n 個の要素 $a_1, \ldots, a_n \in K$ をでたらめに選び，多項式 $a(x) = a_0 + a_1 x + \cdots + a_n x^n$ を考える。さらに $n+1$ 個の要素 $x_0, x_1, \ldots, x_n \in K$ を互いに異なるように選び，対 $\langle x_i, a(x_i)\rangle$ を i 番目の人に与える。$n+1$ 人全員の情報が集まれば，先の Lagrange の補間公式により $a_0, a_1, \ldots, a_n$ が復元されるが，n 人の情報だけからは $a_0, a_1, \ldots, a_n$ の関係は分かっても，a_0 の値を知ることはできない。

この仕組みは，さらに一般することができる。例えば，n 人で秘密を共有するのだが，そのうちの k ($\leqq n$) 人分の情報が集まれば a_0 が分かるという仕組みは次のように作ることができる。$a_1, \ldots, a_{k-1} \in K$ をでたらめに選び，多項式 $a(x) = a_0 + a_1 x + \cdots + a_{k-1} x^n$ を考える。先と同様に n 個の要素 $x_1, \ldots, x_n \in K$ を互いに異なるように選び，対 $\langle x_i, a(x_i)\rangle$ を i 番目の人に与える。k 人分の情報が集まれば，$a_0, a_1, \ldots, a_{k-1}$ が復元されるが，k 人未満の情報だけからは a_0 の値を知ることはできない。

問題 6.7. 秘密の数 a，b，c があり，どれも 97 以下の正の整数だということはわかっている。$f(x) = a + bx + cx^2$ とし，何人かの兄弟それぞれにある特定の数 a に対する $f(a) \mod 97$ を教えた。後日，太郎，次郎，三郎が集まったとき，3 人が知っているのは，$f(1) \mod 97 = 33$ （太郎），$f(2 \mod 97 = 80$ （次郎），$f(3) \mod 97 = 54$ （三郎）だとわかった。秘密の数 秘密の数 a，b，c を特定せよ。

6.2 高速フーリエ変換

本節では複素係数の多項式を考える。

6.2.1 1 の複素 n 乗根と離散フーリエ変換

n を正の整数とする。$\omega_n = \exp(2\pi \mathrm{i}/n) = \cos(2\pi/n) + \mathrm{i}\sin(2\pi/n)$ とすると，1 の複素 n 乗根には，${\omega_n}^k = \exp(2k\pi \mathrm{i}/n)$ $(k = 0, 1, \ldots, n-1)$ があり，それらですべてである。すなわち，$x^n - 1$ は（因数定理により）$(x-1)(x-\omega_n)\cdots(x-{\omega_n}^{n-1})$ と因数分解される。特に ω_n は 1 の原始 n 乗根である。

定義より明らかに ${\omega_{dn}}^{dk} = {\omega_n}^k$ である。特に ${\omega_{2n}}^n = \omega_2 = \exp(\pi \mathrm{i}) = -1$ であり，ゆえに ${\omega_{2n}}^{n+k} = -{\omega_{2n}}^k$ である。

補題 6.8 (総和補題)**.**
任意の正整数 n と整数 $k \in \mathbb{Z}$ について, $\sum_{j=0}^{n-1} {\omega_n}^{kj} = [n \setminus k]n$ である。
証明: $k = l \bmod n$ とすると，$\omega_n^k = \omega_n^l$ だから

$$0 = 1\omega_n^{kn} - 1 = (\omega_n^k - 1)\sum_{j=0}^{n-1} \omega_n^{kj} = (\omega_n^l - 1)\sum_{j=0}^{n-1} \omega_n^{kj}$$

である。従って $k \not\equiv 0 \pmod n$ の場合，$\omega_n^l \neq 1$ だから，$\sum_{j=0}^{n-1} \omega_n^{kj} = 0$ である。また， $k \equiv 0 \pmod n$ ならば $\omega_n^k = \omega_n^l = 1$ だから $\sum_{j=0}^{n-1} \omega_n^{kj} = n$ である □

$f(z)$ を複素関数とするとき，$(f(1), f(\omega_n), \ldots, f({\omega_n}^{n-1}))$ を求めることを（ω_n に関する）$f(x)$ の**離散フーリエ変換**（discrete Fourier transformation，DFT）と呼ぶ。

6.2.2 高速フーリエ変換

$a(x) \in \mathbb{C}[x]$ を n 次未満の多項式とすると，DFT とは，基底 $1, {\omega_n}, \ldots, {\omega_n}^{n-1}$ に関する $a(x)$ の関数値表現を求めることに他ならない。$(a_0, a_1, \ldots, a_{n-1})$ を $a(x)$ の係数表現としよう。

先に述べたように，$a(x)$ の DFT とは，ベクトル $(a_0, a_1, \ldots, a_{n-1})^\top$ に Vandermond 行列

$$V = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \cdots & {\omega_n}^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega_n^{n-1} & \cdots & {\omega_n}^{(n-1)(n-1)} \end{pmatrix}$$

を左から掛けることで得られるが，高速フーリエ変換（fast Fourier transfomation，FFT）という手法を用いれば，計算量 $O(n \lg n)$ で求めることができる。

簡単のため，n は 2^e $(e \in \mathbb{N})$ という形をしているものとする。FFT は，DFT を求める際に $a(x)$ を奇数次項と偶数次項とに分けて，分割統治を行う。すなわち，

$$b(x) = a_0 + a_2 x^1 + \cdots + a_{n-2} x^{n/2-1}$$
$$c(x) = a_1 + a_3 x^1 + \cdots + a_{n-1} x^{n/2-1}$$

とすれば $a(x) = b(x^2) + xc(x^2)$ と書ける。従って，$a(x)$ の DFT は，基底 ${\omega_n}^0 = {\omega_{n/2}}^0, {\omega_n}^2 = {\omega_{n/2}}^1, \ldots, {\omega_n}^{n-2} = {\omega_{n/2}}^{n/2-1}$ に関する $b(x)$ と $c(x)$ の DFT から簡単な計算，すなわち

$$a({\omega_n}^k) = b({\omega_{n/2}}^k) + \omega_n^k c({\omega_{n/2}}^{2k})$$

により求まるので，再帰的に $b(x)$ と $c(x)$ の FFT を行えばよい。また，さらに半周だけずれた点 ${\omega_n}^{n/2+k}$ での値は

$$a({\omega_n}^{n/2+k}) = b({\omega_{n/2}}^k) - \omega_n^k c({\omega_{n/2}}^{2k})$$

で求まるので，$a({\omega_n}^k)$ の計算に使った $b(x)$ と $c(x)$ の DFT は $a({\omega_n}^{n/2+k})$ の計算にもそのまま利用できる。

以上より，FFT の再帰的アルゴリズムは次のようになる。

再帰的 FFT

入力 非負整数 $e \in \mathbb{N}$，n $(= 2^e)$ 次未満多項式 $a(x) \in \mathbb{C}[x]$（の係数表現 $(a_0, a_1, \ldots, a_{n-1}) \in \mathbb{C}^n$）
出力 $a(x)$ の DFT $x = (a(1), \ldots, a({\omega_n}^{n-1}) \in \mathbb{C}^n)$

```
    FFT(e, a(x))
(1) if e = 0 then return a;
(2) b(x) ⟵ (a_0, a_2, ..., a_{n-2});   y ⟵ FFT(e − 1, b(x));
(3) c(x) ⟵ (a_1, a_3, ..., a_{n-1});   z ⟵ FFT(e − 1, c(x));
(4) u ⟵ 1;
(5) for i ∈ [0..n/2 − 1] do
(6)     x_i ⟵ y_i + u z_i;   x_{n/2+i} ⟵ y_i − u z_i;   u ⟵ ω_n u;
(7) return (x_0, x_1, ..., x_{n-1});
```

上のアルゴリズムによって n 次多項式の DFT を計算する場合の計算量を $T(n)$ とすれば，アルゴリズムでは，$n/2$ 次式の FFT を 2 回行って，加減乗計算を n に比例する回数行うだけであるから $T(n) = 2T(n/2) + \Theta(n)$ を満たす。従って $T(n) = \Theta(n \lg n)$ である。

6.2.3 逆変換と多項式の高速乗算法

次に基底 $1, \omega_n, \ldots, {\omega_n}^{n-1}$ に関する関数値表現からの内挿を，FFT を利用して $\Theta(n \lg n)$ で行う方法を検討しよう。$a(x)$ の関数値表現 $\langle 1, y_0 \rangle, \langle \omega_n, y_1 \rangle, \langle {\omega_n}^{n-1}, y_{n-1} \rangle$ が与えられているとする。この関数値表現から $a(x)$ の係数表現を求めることを（ω_n に関する**逆離散フーリエ変換**（inverse Fourie transformation，逆 DFT）と呼ぶ。

$a(x)$ の係数表現は Vandermond 行列

$$V = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \cdots & {\omega_n}^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega_n^{n-1} & \cdots & {\omega_n}^{(n-1)(n-1)} \end{pmatrix}$$

の逆行列 V^{-1} を $(y_0\ y_1\ \ldots\ y_{n-1})^\top$ に左から掛けることで得られるが，

$$W = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega_n^{-1} & \cdots & {\omega_n}^{-n+1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega_n^{-n+1} & \cdots & {\omega_n}^{(-n+1)(-n+1)} \end{pmatrix}$$

とすると，総和補題より $VW = n\mathrm{I}_n$ であることが容易に示されるから，$V^{-1} = \dfrac{1}{n}W$ であることが分かる。W を左から掛ける操作は，ω_n の代わりに ${\omega_n}^{-1} = \exp(-2\pi\mathrm{i}/n)$ を用いるということを除けば，V を掛ける操作，つまり DFT と同じだから，同様に FFT による高速計算が可能であり，その結果を n で割れば内挿が完了する。こうして，$n = 2^e$ の場合に $y = (y_0, y_1, \ldots, y_{n-1})$ からの（FFT による）逆 DFT の計算を $\mathtt{FFT}^{-1}(e, y)$ と記すなら，2 つの多項式の積を求めるアルゴリズムは，次のように書ける。

FFT による複素係数多項式の積の計算

入力 非負整数 $e \in \mathbb{N}$，$n\ (= 2^e)$ 次未満多項式 $a(x) \in \mathbb{C}[x]$ と $b(x) \in \mathbb{C}[x]$（の係数表現）
出力 $a(x)b(x)$ の係数表現 $(c_0, c_1, \ldots, c_{2n-1})$

(1) $(x_0, x_1, \ldots, x_{2n-1}) \longleftarrow \mathtt{FFT}(e+1, a(x))$; $(y_0, y_1, \ldots, y_{2n-1}) \longleftarrow \mathtt{FFT}(e+1, b(x))$;
(2) **for** $i \in [0..2n-1]$ **do** $z_i \longleftarrow x_i y_i$;
(3) $(c_0, c_1, \ldots, c_{2n-1}) \longleftarrow \mathtt{FFT}^{-1}(e+1, (z_0, z_1, \ldots, z_{2n-1}))$;

上記アルゴリズムによる計算量は，FFT，FFT^{-1} による計算がいずれも $\Theta(n \lg n)$ であり，第 2 行目の計算は $\Theta(n)$ だから，全体でも $\Theta(n \lg n)$ である。

問題 6.9. 基底 $1, \omega_4\ (= \mathrm{i}), {\omega_4}^2\ (= -1), {\omega_4}^3\ (= -\mathrm{i})$ に関する $a(x) = 3x^3 + 2x^2 + x$ の DFT を計算せよ。

問題 6.10. $a(x) = 7x^3 - x^2 + x - 10$ と $b(x) = 8x^3 - 6x + 3$ との積を（プログラムまたは手で）FFT を用いて計算せよ。

問題 6.11. 計算量 $\Theta(n \lg n)$ で実行できる $\mathtt{FFT}^{-1}(e, z)$ の再帰的アルゴリズムを書け。すなわち，e を $n = 2^e$ なる非負整数とし，z を（n 次未満の）複素係数多項式 $a(x)$ の DFT とするとき，$a(x)$ の係数表現 $(a_0, a_1, \ldots, a_{n-1})$ を返す。

6.3 整数係数多項式の FFT による積と法計算

FFT を利用した計算法はきわめて効率がよいが，前節のように 1 の複素 n 乗根，特にその浮動小数点表現を用いると誤差が生ずるというデメリットがある。誤差が小さければ，信号処理のような工学的応用の場合は，ほとんど問題がないが，整数係数の多項式の積の場合などのように結果の係数がまた整数になると分かっている場合などでは，この誤差が煩わしいことがある。ほとんどの場合，四捨五入して整数に丸めることで正しい結果が得られるが，この処理の安全性は多項式の次数や係数の大きさに依存するので，誤差解析を行わないと完全には安心して使うことができない.

そこで本節では FFT のアイディアを法計算と組み合わせて，n 次未満の整数係数多項式同士の積を誤差なしで $O(n \lg n)$ で計算する手法について解説する。

定義 6.12 (1 の n 乗根)**.**
n を正の整数とする。$\omega \in K$ の乗法位数 $\mathrm{ord}(\omega)$ が n ならば，ω を 1 の**原始 n 乗根** (primitive n'th root) と呼ぶ。一般に整数 k に対して $\omega^k = 1$ を満たせば，$\omega \in K$ を 1 の k **乗根** (k'th root) と呼ぶ。

問題 6.13. $\omega \in K$ を 1 の n 乗根とするとき，次の条件が互いに同値であることを示せ。

(a) ω は 1 の原始 n 乗根，すなわち $\mathrm{ord}(\omega) = n$ である。
(b) $m \setminus n$ なるすべての m について $\omega^m \neq 1$ である。
(c) n のすべての素因数 p について $\omega^{n/p} \neq 1$ である。

補題 6.14. $\omega \in K$ が 1 の原始 n 乗根のとき，整数 m について ω が 1 の m 乗根になるための必要十分条件は，$n \setminus m$ である。
証明: 定理 3.80 より明らかである。 □

例 6.15.

(a) $K = \mathbb{C}$ とするとき，$\omega = e^{2\pi \mathrm{i}/n} = \cos(2\pi/n) + \mathrm{i}\sin(2\pi/n) \in R$ は 1 の原始 n 乗根である。

(b) フェルマー素数 $2^{2^2} + 1 = 17$ に対しては，0 を除いてすべての $r \in \mathbb{Z}/17\mathbb{Z}$ が 1 の 16 乗根になる。3 は 1 の原始 16 乗根だが，$2^8 - 1 \equiv 0 \pmod{17}$ なので 2 は 1 の原始 16 乗根ではない。

補題 6.16 (総和補題)**.**
$\omega \in K$ を 1 の原始 n 乗根とすると，整数 $k \in \mathbb{Z}$ について，$\sum_{j=0}^{n-1} \omega^{kj} = [n \setminus k]n$ である。
証明: 定理 6.8 と同様である。 □

問題 6.17.
q を素数 p のべき p^e ($e \in \mathbb{N}$) とし，K を位数 q の有限体とする。$q-1$ の素因数分解を $q - 1 = p_1^{e_1} \cdots p_r^{e_r}$ とするとき，次を示せ。

(a) 各 $i \in [1..r]$ について $\mathrm{ord}(b_i) = p_i^{e_i}$ となる $b_i \in K^\times$ が存在する。(ヒント: 方程式 $x^{(q-1)/p_i} = 1$ の解とならない $K^\times$ の要素を考えよ)
(b) 任意の $a, b \in K^\times$ についてて，$\mathrm{ord}(a) \perp \mathrm{ord}(b)$ ならば，$\mathrm{ord}(ab) = \mathrm{ord}(a)\,\mathrm{ord}(b)$ である。
(c) $\mathrm{ord}(b) = q - 1$ となる要素 $b \in K^\times$ が存在する。
(d) $K^\times$ は（乗法に関して）巡回群である。

補題 6.18.
K を位数 q の有限体とする。K が 1 の原始 n 乗根を持つための必要十分条件は n が $q-1$ を割り切ることである。特に p が素数であれば $\mathbb{Z}/p\mathbb{Z}$ は体になるが，$\mathbb{Z}/p\mathbb{Z}$ が 1 の原始 n 乗根を持つための必要十分条件は $n \setminus (p-1)$ である。
証明: 問題 6.17 (c) より $K^\times$ は乗法位数 $q-1$ の要素 b を含む。すると $b^{(q-1)/n}$ は 1 の原始 n 乗根になる。逆に $a \in K^\times$ が 1 の原始 n 乗根であれば，$K^\times$ の要素はすべて 1 の $q-1$ 乗根であるから，補題 6.14 より，$n \setminus (q-1)$ である。 □

問題 6.19.
$(\mathbb{Z}/19\mathbb{Z})^\times$ において，1 の原始 n 乗根が存在するような n をすべて定め，そのような各 n について 1 の原始 n 乗根を 1 つずつ求めよ。

さて，1 の原始 2^e 乗根を持つ体 K 上の多項式環 $K[x]$ においては，$\mathbb{C}[x]$ の場合と同様に FFT が自然に実行可能である。その方法は，必要なすべての演算を $\mathbb{C}$ 上でやる代わりに K で行い，1 の原始複素 2^e 乗根 $\omega_{2^e} = \exp(2\pi\mathrm{i}/2^e)$ の代わりに K における 1 の原始 2^e 乗根 ω を用いるだけですむ。また，その逆元 ω^{-1} を用いることで逆 FFT，すなわち FFT^{-1} も同様に可能である。

さらに，整係数多項式環 $\mathbb{Z}[x]$ 上での FFT を誤差なしで行う方法が上より得られる。つまり，与えられた多項式の係数を，十分大きな素数 p による体 $K = \mathbb{Z}/p\mathbb{Z}$ の要素とみなし，$K[x]$ 上での FFT を行うことで処理するのである。以下にそのアルゴリズムの概要を与える。

法計算 FFT による整係数多項式の積

入力 非負整数 $e \in \mathbb{N}$，$n\ (= 2^e)$ 次未満整係数多項式 $a(x) \in \mathbb{Z}[x]$ $b(x) \in \mathbb{Z}[x]$（の係数表現）
出力 $a(x)b(x)$ の係数表現 $(c_0, c_1, \ldots, c_{2n-1})$

(1) $a(x)b(x)$ のどの係数の絶対値の 2 倍よりも大きい素数 p で $2^{e+1} \setminus (p-1)$ を満たすものを見つける;
(2) $(\mathbb{Z}/p\mathbb{Z})^\times$ における 1 の原始 2^{e+1} 乗根 ω を見つける;
(3) $(x_0, x_1, \ldots, x_{2n-1}) \longleftarrow \mathtt{FFT}(e+1, a)$; $(y_0, y_1, \ldots, y_{2n-1}) \longleftarrow \mathtt{FFT}(e+1, b)$;
(4) **for** $i \in [0..2n-1]$ **do** $z_i \longleftarrow x_i y_i$;
(5) $(c_0, c_1, \ldots, c_{2n-1}) \longleftarrow \mathtt{FFT}^{-1}(e+1, (z_0, z_1, \ldots, z_{2n-1}))$;

(3)–(5) における計算は，すべて p を法として，また，複素根の代わりに (2) で見つけた 1 の原始 2^{e+1} 乗根を用いて実行することは言うまでもない。(1) と (2) に関して補足する。

$a(x)b(x)$ の係数の絶対値の最大は，より小さな見積もりが見つかればそれに越したことはないが，計算量に甚大な影響を与えなければ多少大き過ぎても差し支えないから，

$$M = n \cdot \max\{|a_0|, \ldots, |a_{n-1}|\} \cdot \max\{|b_0|, \ldots, |b_{n-1}|\}$$

くらいで，実用上問題がない。つまり $M < p$ なる素数 p で $2^{e+1} \setminus (p-1)$ を満たすものを見つければ良い。

そのような素数を見つけるには， $m = \lceil M/2^{e+1} \rceil$ から始めて，

$$m2^{e+1}+1, \quad (m+1)2^{e+1}+1, \quad (m+2)2^{e+1}+1, \quad \ldots$$

と順に調べて素数になるものを単に探せばいい。素数かどうか判定する問題は，厳密に行うのは難しいが，素数であるかどうかをほぼ 100 % の確実さで判定する確率的アルゴリズムについて後述する。実用上それで問題がない。

(2) で，$(\mathbb{Z}/p\mathbb{Z})^\times$ における 1 の 2^{e+1} 乗根 ω を見つける必要があるが，そのためには 3.4.5 で扱った -1 の平方根を求める手法がそのまま役に立つ。まず，次の事実に注意しよう。

問題 6.20.
標数が 2 でない体 K における 1 の原始 2^{e+1} 乗根とは -1 の 2^e 乗根に他ならないことを示せ。

従って，$\omega^{2^e} = -1$ となる ω を探せばよい。それは $p-1$ が 2^{e+1} の倍数であるであることに注意すれば，フェルマの小定理より，次のような確率的アルゴリズムで可能である。

1 の 原始 2^{e+1} 乗根

入力 素数 p，非負整数 e （ただし $2^{e+1} \setminus (p-1)$ とする）
出力 1 の原始 2^{e+1} 乗根 ω （すなわち -1 の 2^e 乗根）

root(p, e)
(1) $p-1 = 2^f s$ なる奇数 s と整数 f を求める; （条件より $f > e$ である）
(2) $1 \leqq a < p$ なる整数をランダムに選ぶ;
(3) $b_0 \longleftarrow a^s \bmod p$;
(4) **for** $i \in [1..f-1]$ **do** $b_i \longleftarrow {b_{i-1}}^2 \bmod p$;
(5) **if** $b_i = p-1$ なる i が存在して $i \geqq e$ **then** $\omega \longleftarrow b_{i-e}$ **else** (2) へ戻る;

問題 6.21. 上のアルゴリズムが停止するのは，明らかに (5) で **if** の条件が成り立った場合である。このとき，出力 ω として正しく 1 の原始 2^{e+1} 乗根が得られることを示せ。また，(2) で a がまったく無作為に選ばれるとして，(5) での **if** 条件の成立確率求めよ。