

(5/18) cont'd.

計算量 Computational complexity

算法の効率を測る上での理論的指標へ→

(この授業で扱う)

① 時間計算量 (Time complexity) : 必要な計算のステップ数。
② 空間計算量 (space complexity) : 計算に必要な記憶容量。

(5/18)

↓

③ 計算量の表し方 ... 「漸近記号」 (§1.2)

(5/25)

$$T: \mathbb{N} = \{0, 1, 2, \dots\} \rightarrow \mathbb{R}_{\geq 0}$$

$$\begin{array}{ccc} \psi & & \psi \\ m & \mapsto & T(m) \\ \uparrow & & \end{array}$$

入力データ, "n", ..., 数, 括弧, ワンペース, 多項式, 次數, ...

15

(例) $f(n)$ n : 多倍長整数, ワンペース.

$$f(n) = an, \quad g(n) = bn^2, \quad a > 0, \quad b > 0$$

→ ここで g が f を「遅い」

→ f の「大きさ」。計算量は「漸近的」 n , $f(n)$ と $g(n)$

他の例: $h(n) = c^n$ ($c > 0$), $p(n) = d \log(n)$ ($d > 0$)

等。□

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

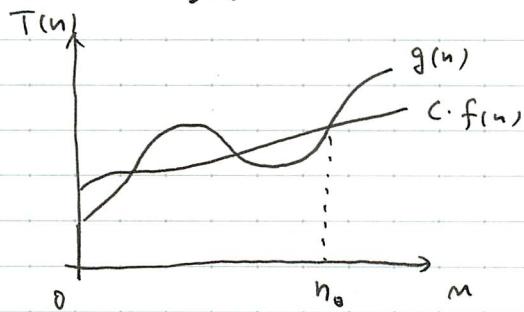
(2) \mathcal{O} -notation

$$g(n) = \mathcal{O}(f(n)) \stackrel{\text{def}}{\Leftrightarrow} \forall c > 0, \exists m_0 > 0 \\ \text{s.t. } n \geq m_0 \Rightarrow 0 \leq g(n) \leq c \cdot f(n) \\ (\Leftrightarrow \lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = 0.)$$

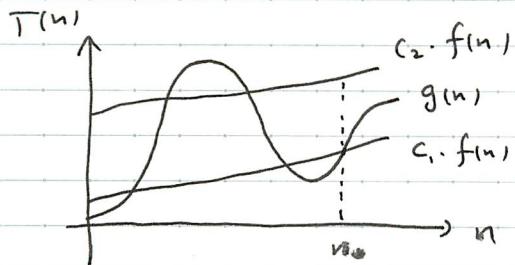
(数学の \mathcal{O} 記法と同一)

(3) Ω -notation : 渐近的下界

$$g(n) = \Omega(f(n)) \stackrel{\text{def}}{\Leftrightarrow} \exists c > 0, \exists m_0 > 0 \\ \text{s.t. } n \geq m_0 \Rightarrow 0 \leq c \cdot f(n) \leq g(n).$$

(4) Θ -notation

$$g(n) = \Theta(f(n)) \stackrel{\text{def}}{\Leftrightarrow} \exists c_1 > 0, \exists c_2 > 0, \exists m_0 > 0 \\ \text{s.t. } n \geq m_0 \Rightarrow 0 \leq c_1 \cdot f(n) \leq g(n) \leq c_2 \cdot f(n).$$

(5) ω -notation

$$g(n) = \omega(f(n)) \stackrel{\text{def}}{\Leftrightarrow} \forall c > 0, \exists m_0 > 0 \\ \text{s.t. } n \geq m_0 \Rightarrow 0 \leq c \cdot f(n) \leq g(n) \\ (\Leftrightarrow \lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = +\infty.)$$

・有効性：有効な（意味のある）計算が行われること。

(例) 有効計算の例：

(整数上、Euclid の互除法)、有理数や無限小数の計算等。

④ アルゴリズム (多倍長整数の加算)

入力：多倍長整数 $a = [a_0, a_1, \dots, a_{k-1}]$, $b = [b_0, b_1, \dots, b_{k-1}]$
出力：" " $c = [c_0, c_1, \dots, c_{k-1}]$ s.t. $c = a + b$.

(1) $\gamma_0 \leftarrow 0$;
 (2) for $i \in [0..k-1]$ do
 $[c_i, \gamma_{i+1}] \leftarrow a_i + b_i + \gamma_i$;
 (3) return $[c_0, c_1, \dots, c_{k-1}]$;

5/8

↓

→ γ は 計算量 を導入 (p. 3-1)

⑤ 多倍長整数の加算の計算量

5/25. 2015

* 計算量のワードごとの四則演算の回数を単位とす (p. 4)

上のアルゴリズムの計算量 (演算回数) を調べると…

(1) $\gamma_0 \leftarrow 0$; 代入は 今日は無視
 (2) for $i \in [0..k-1]$ do ループ k 回
 $[c_i, \gamma_{i+1}] \leftarrow a_i + b_i + \gamma_i$; 加算 2 回
 (3) return $[c_0, c_1, \dots, c_{k-1}]$; return 文は無視

∴ 全体で行われる演算は 加算 $2k$ 回

∴ $T(n) = O(2k)$
 $= O(k)$.

つまり、多倍長整数の加算の計算量は、整数の長さ (k) に比例する。

⑩ 行きつ多倍長整数.

① 2の補数と用いる考え方.

$$n = [a_0 \dots a_{k-1}] = \sum_{i=0}^{k-1} a_i \cdot 2^{n-i}$$

但し、最上位 -1 の a_{k-1} の最上位ビットと行きを表す。3分の3

a_{k-1}

$$n' = \boxed{m_{n-1} | m_{n-2}} \quad | m_1 | m_0$$

$\uparrow 2^{n-2}$ $2^1 2^0$

-2^{n-1}

よし、 $n' = \sum_{j=0}^{n-2} m_j 2^j + (-1)m_{n-1} \cdot 2^{n-1}$.

したがって $-2^{n-1} \leq n' < 2^{n-1}$.

よし、 n 全体について $-2^{nk-1} \leq n < 2^{nk-1}$.
の範囲で。

問題2.7

行きつ多倍長整数の減算と、10-1=m<₂と、整数の減算の組合せで表す。

基本的に方法は多倍長整数の計算と同じである。
最上位ワードの最上位ビットと行きの用いる
118の2。この部分は計算上必要ない。この手ノカ
以降。



2.2 多項式の表現

2.2.1 多項式の表現と加算

• R : 可換環.

• $R[x, y, \dots, z]$ 变数 $x, y, z \in R$, R 上の多項式全般.

($R[x, y, \dots, z]$ 上の自然な n への乗算が定義され,
 $R[x, y, \dots, z]$ は可換環である).

• 当面の変数を x : 主變数.
 係数を a : $\mathbb{Z}[x]$.

Def (主項, 主係数, 次数, $\tau = \gamma$)

$$a(x) = a_n x^n + \dots + a_1 x + a_0 \in \mathbb{Z}[x], a_n \neq 0$$

$a_n x^n$: $a(x)$ の主項 (the leading term), $lt(a(x))$.

a_n : --- の主係数 (the leading coefficient), $lc(a(x))$.

n : --- の次数 (degree), $\deg(a(x))$.

$a(x) \neq 0 \Rightarrow \tau = \gamma$ (monic) $\Leftrightarrow lc(a(x)) = 1$.

回

Def (1変数多項式の表現)

$$a(x) = a_n x^n + \dots + a_1 x + a_0 \in \mathbb{Z}[x], a_n \neq 0$$

と

$$(a_0, \dots, a_n) \in \mathbb{Z}^{n+1} \sim \mathbb{Z}^{\gamma}.$$

57/25 回下

④ 1変数多項式の加算.

$$a(x) = \sum_{i=0}^k a_i x^i = (a_0, a_1, \dots, a_k),$$

$$b(x) = \sum_{i=0}^k b_i x^i = (b_0, b_1, \dots, b_k)$$

6/1

など.

$$c(x) = a(x) + b(x) = \sum_{i=0}^k c_i x^i = (c_0, c_1, \dots, c_k)$$

となる.