

計算機ソフトウェア 第一回

電気電子工学科
黒橋禎夫

アルゴリズムと計算量

- アルゴリズム

与えられたデータから：「入力」

目的の情報を：「出力」

見つけ出す or 作り出す：「手続き」

最近は一般語になりつつありますね！

アルゴリズムと計算量

- アルゴリズム
 - 一種のモデル化
 - 抽象的な問題を正確に表現
- 計算量
 - アルゴリズムの「良さ」を表す専門用語
 - (「評価」=工学の基本)

アルゴリズムの記述

例. 「自然数, 3, 8, 16, 4, 9, の最大値を求める」

- 言葉をいっぱい使う

「順番に大きなものを覚えておく」

曖昧性

複雑さの議論できない

- 計算機がわかるように記述

アルゴリズムの記述

- ある程度抽象化

(厳密さと簡潔さの)バランス感覚です

入力: 正の整数値 n と n 個の実数値

出力: x_1, x_2, \dots, x_n 中の最大値

手続き:

1. $y \leftarrow x_1$
2. for i 2 until n do
 - if $x_i > y$ then $y \leftarrow x_i$
3. return y

演習 手続きを記述せよ

- 入力: 正の整数値 n と,
 n 個の実数値 x_1, x_2, \dots, x_n と, y
- 出力: x_1, x_2, \dots, x_n の中で最も y に近いもの

計算量

- 入力の大きさ n に対してどれくらいの計算が必要か
- 「入力の大きさ n 」 = 問題の大きさ
- 少し抽象化する(オーダー)

オーダーの定義

$f(n)$ と $p(n)$ を自然数の上で定義された関数とする
任意の n に対して

$$f(n) / p(n) < C$$

をみたす n によらない定数 C が存在するとき

$$f(n) = O(p(n))$$

という

この定義は理解しておいてほしい！

オーダーの意味

- $p(n)$ はできるだけ簡単なもので議論すればよい
議論する = 最悪の場合を考える

例. $f(n) = 3n^2 + n$ に対して

$C = 3$ とか $C = 4$ を考えれば

$p(n) = n^2$ で $f(n)$ を抑えられる

多項式だと一番大きな(次数の)項をとればいい！

演習 オーダを示せ

1. $f(n) = 5n \log n + 3n^2$

答え $O(n^2)$

2. $f(n) = 2^n + 5n^5$

答え $O(2^n)$

「最大値を求めるアルゴリズム」の場合

1. $y \leftarrow x_1$

代入 a秒

2. for i 2 until n do

 if $x_i > y$ then $y \leftarrow x_i$

比較 (n-1)b秒

代入 (n-1)c秒

3. return y

d秒

$$f(n) = (n-1)(b+c) + a + d$$

よって計算量は $O(n)$

計算量の感覚

	n	1	1,000	1,000,000
$O(1)$	理想的	1 ms	1 ms	1 ms
$O(\log n)$	◎	1 ms	7 ms	14 ms
$O(n)$	○	1 ms	1 sec	17 min
$O(n \log n)$	○	1 ms	7 sec	4 hours
$O(n^2)$	△	1 ms	17 min	30 years
$O(n^3)$	△ —			
$O(cn)$	×			

計算量 補足

- 本当は複雑度、計算量は二種類ある
 - 空間複雑度 (space complexity)
 - cpuの数やメモリの量による制約があるため
普段は意識しない(=そんなプログラム書かない)
 - 時間複雑度 (time complexity)
 - だから基本的にはこちらを考えます
- 一応、両方あるってことを覚えといてください

本講義の位置付け

プログラミング演習：実際のインプリメント

本講義：抽象化して「良さ」や「設計」を扱う