講義の概要

アルゴリズム論

中野

Note 1　講義の概要

2020.4.7作成　2021.4.6update

授業のＨＰ

http://www.cs.gunma-u.ac.jp/~nakano/Algo/index.html

講義の概要

計算機で問題を解くためには、
(1) 問題を数学的に定式化し、
(2) 問題を解く手順(アルゴリズム)を設計し、
(3) この手順を計算機が処理可能な形式に翻訳したもの(プログラム)を作成、
しなくてはなりません。

(下記文献Great Ideas in Computer Science, p43より)

計算機科学とは、アルゴリズムの研究ならびに、計算機によるその自動的実行の学問ともいえる。

The field of computer science maybe thought of as the study of algorithms and their automatic execution by machines.

この講義では、アルゴリズムの設計と評価について解説します。
幾何アルゴリズムも紹介します.

アルゴリズムとは

アルゴリズムとは、
問題を解くための手順である。アルゴリズムには、厳密に定義された入力と出力がある。
アルゴリズムは出力を求めるための手順をキチンと定義したものである。

問題とは
本講義においては、問題(a problem)とは、通常いくつかのパラメータを含むような、
一般的な質問であるとしよう。

例
a万円を利息b%で複利でc年間借りると、元利合計はいくらになるか?
整数aの平方根の値を小数点以下bケタまで求めよ。
2020年a月b日の朝7時に桐生駅を出発するとき、広島駅まで最短の時間でいくための
列車の選びかたは?
2020年a月b日の朝日新聞群馬版の記事のみだしの一覧を作成せよ。

アルゴリズムとは（つづき）

問題でない例

本aはおもしろいか。　("おもしろい"の定義はアイマイ。)
今朝の新聞を要約せよ。(要約の定義がアイマイ。)

問題の全てのパラメータに具体的な値を設定したものをインスタンス(an instance)という。

インスタンスの例
300万円を年利息4%で複利で25年間借りると、元利合計はいくらになるか?

アルゴリズムとは（つづき）

解くとは

アルゴリズムAが、問題Bのすべてのインスタンスに適用でき、
各インスタンスCに対して常に正しい解を求めるとき、
アルゴリズムAは問題Bを解くという。

解法の比較

次のインスタンスを解く、ふたつの解法A,Bを比較しよう。
300万円を年利息4%で複利で25年間借りると、元利合計はいくらになるか?(ちなみに答は7997509円)

解法A

3000000 x 1.04 x 1.04 x 1.04 x ... x 1.04 =
25回乗算

乗算25回

解法B

$a1 = (1.04)$
$a2 = (a1)^2$
$a3 = (a2)^2 = (1.04)^4$
$a4 = (a3)^2 = (1.04)^8$
$a5 = (a4)^2 = (1.04)^{16}$
3000000 x a1 x a4 x a5 =
(25を2進法で表記すると11001)

乗算7回

アルゴリズムの比較

問題　a万円を年利息b%で複利でc年間借りると、元利合計はいくらになるか?

解法Aに基づくアルゴリズムA

1.整数a,b,cを入力

2.temp = a x 10000

3.ratio=1 + 0.01 x b

4.for i = 1 to c　　temp = temp x ratio

5.tempを出力

解法Bに基づくアルゴリズムB

1. 整数a,b,cを入力

2. temp = a x 10000

3. ratio=1 + 0.01 x b

4. year = c

5. while year<>0 do

5-1　　if year は2で割りきれない

then temp = temp x ratio

5-2　　year は year/2の切り捨て とする

5-3　　ratio = ratio * ratio

6.tempを出力

	乗算の回数	記憶領域
アルゴリズムA	ちょうど2+c	3
アルゴリズムB	高々2+2 log c	3

アルゴリズムの評価

アルゴリズムの評価は
計算時間(time complexity)と記憶領域(space complexity)
で行なうことが多い。
アルゴリズムの計算時間とは、各入力サイズに対して、そのサイズの任意のインスタンスを解くのに必要な最大の演算回数を返す関数である。

(計算時間は関数であることに注意。最悪の場合で評価していることに注意。)
(演算の合計回数のみチェックします。各演算が何回か等の詳細は気にしません。)

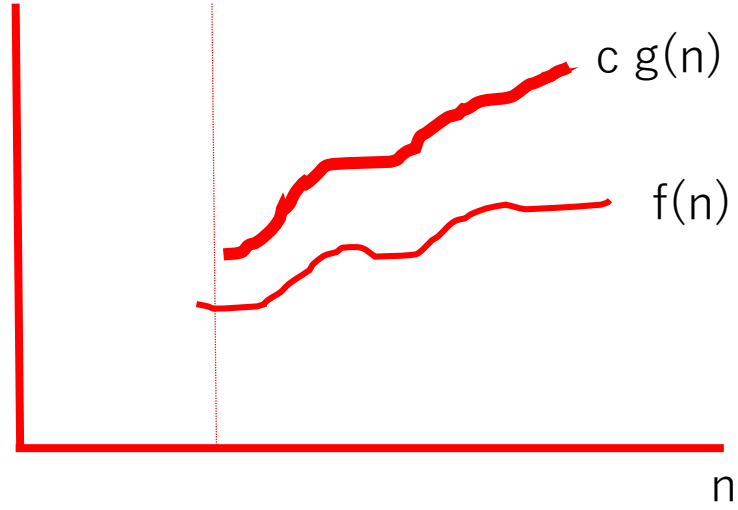
入力サイズ

入力サイズ(の厳密な定義)は入力を表現するのに必要な総bit数である。
例えば、一個の整数cが入力ならば、入力サイズは log cである。
(厳密には、 log (c+1) の切り上げ)

問題によっては、(厳密ではないが、お手軽に)、他を用いてもよい。
ソートの時はデータの個数、グラフに関する問題では点の個数と辺の本数を用いるのが自然である。

アルゴリズムBの入力サイズは log a + log b + log c
アルゴリズムBの乗算の回数は高々 2 + 2 log c 回
アルゴリズムBの記憶領域は log a + log b + log c

O-表記(オーダー表記)



O(g(n))は関数の集合である。
下記のように定義される。
$O(g(n)) = \{ f(n)$: there exist positive constants
c and n_0 such that
$0 \leqq f(n) \leqq c\ g(n)$ for all $n \geqq n_0\}$

例
1000n+105 は O(n log n)に含まれるかな?

例
$O(n^3)$ は要素として n^2 や $n^2 \log n$ を含む
入力グラフの点の数がn, 辺の数がmのとき、
ヒープを用いたダイキストラ法の計算時間はO(m lon n)である。

計算機で 解ける問題 と 解けない問題

計算機で、ある問題が解けないとは、いろいろな意味がある。

(1)計算機に日本語を理解させる
　数学的な定式化がわからない
(2)Pascalでリアルタイムゲームを作る
　ある言語ではタイミングや割り込みを扱えないのでダメ(システムコールが必要)
(3)300ケタまでの素数の表を作成する
　メモリやディスクが足りなくて解けない
(4)巡回セールスマン問題の解を求める
　時間が非常に非常に非常に長くかかる
(5)計算機プログラムの停止問題
　計算不能

チューリングマシンを用いても、問題には、
計算可能(computableまたは可解solvable)なものと、
計算不能(noncomputableまたは非可解unsolvable)なもの
の2種類がある。
今後、どのように計算機が高速化しようとも、計算不能とよばれる
一連の問題があり、これを解くアルゴリズムは存在しないのである。有名な計算不能問題に、
“プログラムの停止性問題(The halting problem)[Turing,1936]"
等がある。

計算可能な問題の4種類

(1)多項式時間アルゴリズム(計算時間が多項式のオーダーのアルゴリズム)が
知られている問題 (クラス P)
ソート・サーチ・グラフの最短経路・平面グラフの判定・最小木・
線形計画法[Karmarkar,1984]・.....
(2)指数関数時間かかることが判明しているもの
グラフの全ての部分木の発生・....
(3) 解の検証は多項式時間でできるが、
解を求める多項式時間アルゴリズムは知られていないもの
(クラス NP完全 (NP-Complete))
充足可能性問題・独立点集合問題・ハミルトン閉路問題・.....
(4)上記3つのどれにはいるか、現在不明のもの
グラフ同形問題(graph isomorphism)・素因数分解(prime factoring)・...

チャーチ - マルコフ - チューリングの命題 (Church-Markov-Turing thesis)(1930年代)

定性的解釈

Any nontrivial computer language that one can invent is apparently capable of computing no more and no fewer functions than all the other nontrivial programming languages. (Creat Ideas in Computer Science, p187)

（問題が計算可能であるか計算不能であるかは、計算機モデルに依存しない。）

定量的解釈

Any resonable attempt to model mathematically computer algorithms and their time performance is bound to end up with a model of computation and associated time cost that is equivalent to Turing machines within a polynomial. (Computational complexity, p36)

（アルゴリズムが、多項式時間であるか指数関数時間であるかは、
計算機モデルに依存しない。）

以上の理由により、
　ある言語のプログラムで解ける問題　＝ 計算可能(計算機で解ける)
　ある言語のプログラムで解けない問題 ＝ 計算不能(計算機で解けない)
と考えてよい。
インスタンスを解くのではなく、問題を解くことに注意。
また、
　ある言語のプログラムで
多項式時間で解ける問題 ＝ クラス Pに属している問題
と考えてよい。

　この講義では、高速な多項式時間アルゴリズムの設計と評価について解説します。