計算情報学 I

名古屋大学　情報文化学部
自然情報学科３年
第３回

鈴木泰博
情報文化学部・大学院情報科学研究科
複雑系科学専攻

Science of interactions

Agenda

計算情報学　Ｉ　第３回

アルゴリズムのお話（理論計算機科学入門）

１．前回の復習
２．アルゴリズムとは何か
３．計算可能性
４．計算量理論

復習のための演習問題①

1. 10進法の5を２進法で表せ。
2. ２進法の1000を10進法で表せ。
3. 1/5を2進法で計算せよ
4. 2進法で正確に表せる（無限小数にならない）数を挙げよ。

復習のための演習問題①
解答

1. 10進法の5を２進法で表せ？

   ans. 0, 1, 10, 11, 100, 101

1. ２進法の1000を10進法で表せ。

   ans. 8

1. 1/5を2進法で計算せよ

   ans. 0.0011001100…

2. 2進法で正確に表せる（無限小数にならない）数を挙げよ。

   ans. 2^{n},（n=0,1,2…）

Science of interactions

復習のための演習問題②
解答

1. 以下の数値を有効数字3桁になるように四捨五入で丸めよ

   1.2345　135.79　0.00005438

2. 10進法で仮数部4桁、かつ切り捨てによる丸めを行うとして以下を計算せよ。

   $A=0.1234 \times 10^3$, $B=0.1056 \times 10^1$,

   $C=0.1072 \times 10^1$ としたときに、

   ① $(A+B)+C$　　②$A+(B+C)$

   ans. $(A+B)+C = 0.1254 \times 10^3$, $A+(B+C)=0.1255 \times 10^3$

3. 10進表で仮数部5桁、かつ切り捨てによる丸めを行うとして以下を計算した場合の有効数字の桁落ちは何桁か

   0.0001234555 – 0.000123455

   ans. 4桁

要約

- 2進法の表現、有限性、離散性（連続は扱えない）ことが計算機のクセ
- この“クセ”が誤差をもたらす
  - 和算は値の小さい順に行う（演習問題②－２）
  - 小さな値同士の減算は避ける(演習問題②－３）
- この演習問題では取り上げなかったが、打ち切り誤差も有限性の問題から出てくる。

Science of interactions

ニュートンとライプニッツ

“神”の存在証明をいかに行うか？

ニュートン
神の見えざる手

物理学の系譜
（“力を隠す”様式の探求）

現代数学に影響を
与える

ビショップ
直観主義数学

アルゴリズム検証
の数学的証明（代数
的仕様記述）

ライプニッツ
アルゴリズム的に神を表現

バベッジ
階差計計算機 (ADA)

チューリング
計算の数学理論

電子計算機(ENIAC)

新しい計算モデル（分子、量子）

アルゴリズム＝手続き

語源：9世紀頃にアラビアで算術の教科書を執筆した
Abu Ja'far Mohammed ibn Musaal-Kowarizumi
（フワーリズム(旧ソ連）の出身者という意味）

“神の見えざる手”ではなく、問題を解く方法を“あからさまに”すべて記述すること。
（”あからさまに“＝背理法などは使わずに）

$$1 + 2 + 3 + 4 + 5 + 6 + 7 \ldots + 100$$

$$\updownarrow$$

$$(100 + 1) \times 50 = 5050$$

数が大きくなったときの加算の回数は？

最大公約数

問題　24と60の最大公約数を求めよ。

2 | 24　60
2 | 12　30　→　12
3 | 　6　15
　　 2　 5

ユークリッドの互除法
① n を m で割り、その余りを r とする。
② r = 0 ならば m が最大公約数
③ r ≠0ならば n を m, r を m して ①へ

60 / 24 = 2 余り 12
24 / 12 = 2 余り 0
最大公約数は12

アルゴリズムの証明

補題1　$a > b > 0$, aを bで割った商をr（$r>0$）とする。a,bの最大公約数をcとするとrはcで割り切れる。

補題2 bとrの最大公約数c'でaは割り切れる。

命題　アルゴリズムの正当性
$a > b$ よりこのアルゴリズムの手続きはいずれ停止する。
補題１，２より、$c < c'$とすると、aはaとbの最大公約数よりも大きな最大公約数を持つことになり矛盾する。よって$c = c'$

アルゴリズムの数学的定義

チューリングマシン(A.Turing,1912-54)
“計算”の数学的定義を与える

RAMモデル、λ計算、帰納関数、whileプログラミング, e.t.c
#ちなみにARMS, P systemsも
すべて計算能力はチューリングマシンと等価。

“チューリングマシンは実行可能な手続きという概念に対応し得る十分に汎用性のある形式的な計算モデルである“
(チャーチの提唱):
つまり、チューリングマシンで表現される手続きをアルゴリズムの数学的な定義としよう。

Science of interactions

要約

- アルゴリズム＝手続き
- アルゴリズムの数学的定義はチューリングマシンによる。

アルゴリズムの評価

入力のサイズが大きくなるに従って、アルゴリズムの時間効率がどのように変化するか？

評価のための単位時間には、
CPU, ALUの性能
コンパイラの特性
２次記憶、入出力処理のオーバーヘッド
プログラミング言語の特性
…
はすべて“無視”する。そして、
プログラムの“１命令”＝１単位時間
とみなす。

アルゴリズムの評価

入力サイズnに対する実行時間f(n)の極限での振舞い(計算量)に注目する。

そのため、f(n)の係数や低次の項はすべて無視する。f (n)の高次の項をアルゴリズムのオーダーをよぶ。

オーダーの違いによる計算量の違い

Science of interactions

オーダーによる計算量の違い

	A(n)	B(n^2)	C(2^n)
10	10^{-7}	10^{-6}	10^{-5}
20	2×10^{-7}	4×10^{-6}	10^{-2}
30	3×10^{-7}	9×10^{-6}	10
50	5×10^{-7}	2.5×10^{-5}	10^{7}=130日
100	10^{-6}	10^{-4}	4×10^{14}年

毎秒１億回の演算を実行する計算機の場合

演習
以下の計算量を評価せよ

① $2n+100$

② $5n\log n + 2n^2$

③ 2^n+n^{100}

解答

① n

② n^2

③ 2^n

計算量の計算
演習

次のアルゴリズムの計算量を見積もれ

```
for i = 1 to n
   {
    a:=(1+2)x4
   }
```

```
for i = 1 to n
   {
       b:=a+b
       for j=1 to n
             {
            a:=(1+2)x4
            }
   }
```

計算量の計算
解答

次のアルゴリズムの計算量を見積もれ

```
for i = 1 to n
   {
    a:=(1+2)x4
   }
```

Ans. $O(n)$

```
for i = 1 to n
   {
        b:=a+b
        for j=1 to n
               {
               a:=(1+2)x4
               }
   }
```

Ans. $O(n^2)$

Science of interactions

計算可能性

- 停止問題(アルゴリズムが停止するか判別する問題)
- ヒルベルトの第10問題
  “n 個の未知数を含む整数係数の多項式 P(x1,x2,..., xn) に対し，方程式 P(x1,x2,...,xn) = 0（ディオファントス方程式または不定方程式と呼ぶ）が整数解を持つか否かを有限的に判定する方法をみつけよ”

  などなどはチューリングマシンで計算できない＝アルゴリズムが存在しないことが証明されている。

このような問題を決定不能な問題とよぶ

つまり、アルゴリズムは存在しない。

Science of interactions

計算ができないということ

演習

1. クレタ人は嘘つきだとクレタ人が言った。クレタ人は果たして嘘つきであるか、そうでないかを判定せよ。

2. “このスクリーンに書いてあることは、すべて嘘です”ではこの講義の内容はすべて嘘であるのか判定せよ。

自己言及のパラドクス

Science of interactions

停止問題

準備①：halt(p, x)は任意のプログラムpに対し入力xがあった場合にプログラムが停止するならばT、停止しない場合はFを返す。

準備②：haltを用いてプログラムの停止問題のプログラムϕをつくる。Φはもしhalt(p,x)が停止するならF, 停止しないならTを返す。

もし、halt (ϕ　ϕ)は停止すると仮定すると、halt(p,x)、この場合は定義よりpは任意のプログラムで、今はp=x=ϕなのでhalt(ϕ,ϕ）、は停止しないことになる。逆にhalt(ϕ　　ϕ)が停止しないと仮定すると、halt(p,x)=halt(ϕ,ϕ）は停止することになる。よって矛盾する。

計算量
計算の複雑さの理論

前提：アルゴリズムが“ある”問題において、そのアルゴリズムがどの程度複雑かを評価する理論

- 時間計算量
    - 解くのに必要な時間
- 領域計算量
    - 解くのに必要な記憶領域

P（多項式時間で答えが出る）、NP（Pではない場合）

NP困難：NPのクラスの問題だが、Pのクラスに多項式時間で帰着可能
NP完全：NPのクラスの問題で、Pに帰着できない場合（最も難しい問題）

（NP完全問題の例　→　SAT問題）

Science of interactions

ＮＰ完全の近似解法

- 欲張り法
  - 各時点で最適な手法を選択する。単純だが強力な方法
- ランダム法
  - 乱数を用いてランダムに解の生成を行い、その中から最適解を作り出す。
- 緩和法
  - まず、条件を緩和した問題を考え、しだいに条件を厳しくしていく方法
- 逐次探索法
  - まず、適当な近似解を求めその近傍の最適解を探索する。
- アニーリング法
  - 逐次近似法で近傍を確率的に求める
- 遺伝的アルゴリズム
  - ランダム法を並列に行う。
- Ａアルゴリズム
  - グラフ探索問題において、現在地点からゴールまでの距離を推定しより短い頂点に移る。この際にその距離が「真の距離よりも大きくない」ことが保障されている場合、Ａ＊アルゴリズムとよばれる。

例）TSP

＃量子計算、分子計算、膜(化学）計算。。ちょっとインチキっぽいけど。