

大阪電気通信大学 情報通信工学部 光システム工学科 2年次配当科目

コンピュータアルゴリズム アルゴリズムとは何か？

第1講: 平成20年10月3日 (金) 4限 E352教室

中村 嘉隆(なかむら よしたか)
奈良先端科学技術大学院大学 助教
y-nakamr@is.naist.jp
<http://narayama.naist.jp/~y-nakamr/>

講師紹介

- ▶ 中村 嘉隆(なかむら よしたか)
 - ▶ 奈良先端科学技術大学院大学 情報科学研究科 情報基礎学講座 助教
- ▶ 連絡方法
 - ▶ 直接, 奈良先端科学技術大学院大学 情報科学研究科 B棟 5階 B504へ来る(生駒市)
 - ▶ 電子メールを y-nakamr@is.naist.jpへ
- ▶ 講師歴
 - ▶ 2年目
(昨年, ここで教えたのが初めての経験)

2008/10/3

第1講 アルゴリズムとは何か？

2

授業が分からなかったら

- ▶ 授業後に質問
- ▶ ミニレポートの質問欄で質問
- ▶ メールで質問
 - ▶ メールアドレスは y-nakamr@is.naist.jp
 - ▶ 題名と差出人は明確に
 - ▶ 学籍番号, 氏名, 質問の具体的な内容をお願いします
 - × 題名: 質問, 本文: 今日の授業が分かりません
 - ▶ 授業に出てない人(ミニレポートを出していない人)への返事は後回しになります

2008/10/3

第1講 アルゴリズムとは何か？

3

成績のつけ方

- ▶ 試験
 - ▶ 試験で合格点を満たせば無条件で合格
- ▶ 授業ごとにミニレポート
 - ▶ 授業終了後に提出, 次の授業で解説
 - ▶ 試験の持ち込みはこのミニレポートのみ
 - ▶ 試験で成績が悪くてもミニレポートの提出実績を加味して救済
 - ▶ 一週遅れるごとに提出点は半減
 - ▶ 一ヶ月前のミニレポートは出しても提出点になりません
 - ▶ <http://narayama.naist.jp/~y-nakamr/lecture/computeralgorithm2008/index.html>
- ▶ 基本的に出席はとりません

2008/10/3

第1講 アルゴリズムとは何か？

4

この授業は…

1. アルゴリズムの基本
 - ▶ アルゴリズムとは
 - ▶ データ構造の基本
2. 探索アルゴリズム
 - ▶ 目的のデータの見つけ方
3. ソーティングアルゴリズム
 - ▶ データの並べ替え

2008/10/3

第1講 アルゴリズムとは何か？

5

教材

- ▶ 講義資料
 - ▶ <http://narayama.naist.jp/~y-nakamr/lecture/computeralgorithm2008/index.html>
- ▶ アルゴリズムとデータ構造
 - ▶ 石畑 清 著, 岩波書店
 - ▶ ISBN4-00-010343-1, 3920円, 1989年
- ▶ コンピュータサイエンス入門
アルゴリズムとプログラミング言語
 - ▶ 大堀 淳, ジャック ガリグ, 西村 進 著, 岩波書店
 - ▶ ISBN4-00-005006-0, 2500円, 1999年
- ▶ アルゴリズム理論入門
 - ▶ 岩間 一雄 著, 昭晃堂
 - ▶ ISBN4-7856-3125-2, 3300円, 2001年

2008/10/3

第1講 アルゴリズムとは何か？

6

アルゴリズムとは何か

アルゴリズム

- 特定の目的を行う(問題を解く)ための手順
- 入力と出力があり、一般的な用語で記述される
- ≠プログラム
 - プログラムはそれをコンピュータに実行させるため、コンピュータが理解できる具体的な命令で書いたもの

アルゴリズム (algorithm)

- 〔補説〕アラビアの数学者アル＝フワズミの名にちなむ
- 〔1〕もとは算用数字を用いた筆算のこと。
- 〔2〕計算や問題を解決するための手順、方式。特にコンピューター・プログラムの適用可能な手続きをいうことが多い。 [三省堂 大辞林より]

2008/10/3

第1講 アルゴリズムとは何か?

Page 7

例えば、連絡網を回したい

➤1 クラス 80 人とする、電話は 1 回 5 分とする

〔方法 1〕

- 担任の先生が出席番号 1 番の人に「次の出席番号の人に掛けることを繰り返してもらって」と頼む
- アルゴリズム
 - 出席番号 1 の人から始める(i を 1 とする)
 - 出席番号が i の人は i + 1 の人に電話をかける
 - i が 80 になると終わり
- 必要な時間
 - 5 分 × 80 回 = 400 分 = 6 時間 40 分

2008/10/3

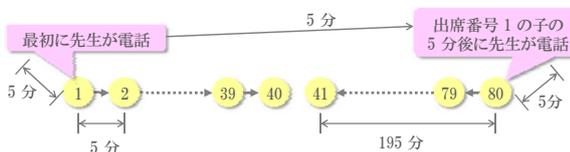
第1講 アルゴリズムとは何か?

8

例えば、連絡網を回したい

〔方法 2〕

- 担任の先生が出席番号 1 番の人と 80 番の人に電話を掛ける
- 必要な時間
 - 5 分 + 5 分 × 40 回 = 205 分 = 3 時間 25 分



2008/10/3

第1講 アルゴリズムとは何か?

9

例えば、連絡網を回したい

〔方法 3〕

- 担任の先生が出席番号下一桁が 1 の人に電話を掛ける
- 必要な時間
 - 35 分 + 5 分 × 10 回 = 85 分 = 1 時間 25 分



2008/10/3

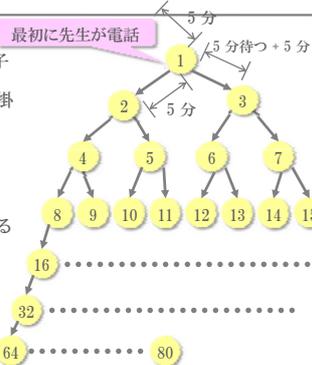
第1講 アルゴリズムとは何か?

10

例えば、連絡網を回したい

〔方法 4〕

- 担任の先生が出席番号 1 の人に掛け、各学生は自分の出席番号の倍の子とその次の子に掛ける
- アルゴリズム
 - 出席番号 1 の人から始める (i を 1 とする)
 - 出席番号が i の人は 2i と 2i + 1 の二人に電話を掛ける
 - i が 80 以上になると終わり
- 必要な時間
 - 5 分 + 10 分 × 6 回 = 65 分 = 1 時間 5 分



2008/10/3

第1講 アルゴリズムとは何か?

11

もっと人数が多いと...

➤ 例えば1000人

- 方法 3
 - 10 人 毎ではなく、50 人 毎にする
 - 95 分 + 5 分 × 50 回 = 345 分 = **5 時間 45 分**



2008/10/3

第1講 アルゴリズムとは何か?

12

もっと人数が多いと...

最初に先生が電話

- 方法 4
 - 1 段増える毎に伝わる人は 2 倍
 - 5 分で 1 人
 - 5 分 + 10 分で 3 人
 - 5 分 + 10 分 × 2 で 7 人
 - ...
 - 5 分 + 10 分 × 8 で 511 人
 - 5 分 + 10 分 × 9 で 1000 人超
 - 95 分 = **1 時間 35 分**
 - 2000 人でも 1 時間 45 分

2008/10/3 第1講 アルゴリズムとは何か? 13

アルゴリズムをなぜ勉強するのか

- 連絡網の例
 - 80 人で方法 1 だと約 6 時間, 方法 4 だと約 1 時間
 - 1000 人だと方法 1 で約 3 日 11 時間, 方法 4 で約 1 時間半
- 単純に考えるとできないことでも, 工夫すればできるようになるかも
 - みんなができないことができる
 - 世の中の役に立つ(お金儲け?)
 - 時間が有効に使え
 - 仕事をする時間を減らして趣味に使え

→ 人生を上手に過ごすことに使える

2008/10/3 第1講 アルゴリズムとは何か? 14

ハードウェアの進化 vs. アルゴリズムの進化

- 初めて線型計画法というアルゴリズムがコンピュータ上で解かれた 1987 年に比べて, 2002 年は **190 万倍** 早く解けるようになった(米ライス大 Bixby 教授)
- その間のハードウェアの進化は **約 800 倍**
- アルゴリズムの進化は **約 2400 倍!**

2008/10/3 第1講 アルゴリズムとは何か? 15

アルゴリズムの定義

- 入力と出力があり, 処理の手続きは一般的な用語で表現される
- 以下の 4 つの性質を持たなければならない
 - 正当性
 - 導き出した結果(出力)は正しい
 - 決定性
 - 同じ入力に対しては, 同じ出力を返す
 - 有限性
 - 各手続きは有限である
 - 無限に繰り返す手続きがあってはならない
 - 停止性
 - 任意の入力に対しては, 必ず停止する

2008/10/3 第1講 アルゴリズムとは何か? 16

講義で対象とする計算モデル

- 逐次型の計算機(コンピュータ)
 - 1 つの中央演算装置(CPU)と RAM(Random Access Memory)を持つ
 - RAM には番地があり, CPU は番地を指定することで任意のアドレスのデータを読み書きできる

- 並列コンピュータはここでは考えない

2008/10/3 第1講 アルゴリズムとは何か? 17

アルゴリズムとプログラム

- アルゴリズムは記述言語には依存しない
- 計算機にはプログラム言語として与える
 - C, Pascal, Java など

```

int gcd(int m, n) {
    int r;
    for (;;) {
        r = m % n;
        if (r == 0) break;
        m = n; n = r;
    }
    return (n);
}
    
```

- アルゴリズムは**抽象的な概念**であり, プログラムはそれを**具体的な形にしたもの**

2008/10/3 第1講 アルゴリズムとは何か? 18

アルゴリズムを設計するために

- 問題を明確に定義
 - どのような仮定の下で考えるか
 - どのような情報が与えられているか
 - 求めるものは何か
 - 問題の解をどのように認識するのか
- 問題を数理的なモデルで表現
- 制御構造とデータ構造でアルゴリズムを設計
 - 制御構造
 - if-else, while, for, return, goto文
 - データ構造
 - 配列, リスト, キュー, スタック, 木

2008/10/3

第1講 アルゴリズムとは何か?

19

アルゴリズムの記法

- 自然言語表現
 - 日本語などで簡潔に表現したもの
- 図形表現
 - フローチャート(Flow Chart: 流れ図)
 - PAD(Problem Analysis Diagram)
- 擬似言語表現
 - C, Pascal などの高級言語の制御構造と自然言語を組み合わせた表現方法

2008/10/3

第1講 アルゴリズムとは何か?

20

擬似言語とフローチャート表現(1)

➤ 代入文

変数 ← 式 ;



➤ ブロック

“{”, “}” で囲まれた部分



2008/10/3

第1講 アルゴリズムとは何か?

21

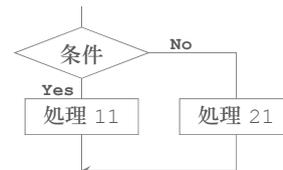
擬似言語とフローチャート表現(2)

➤ if 文

```

if (条件) {
  処理 11 ;
  処理 12 ;
  :
}
else {
  処理 21 ;
  処理 22 ;
  :
}

```



2008/10/3

第1講 アルゴリズムとは何か?

22

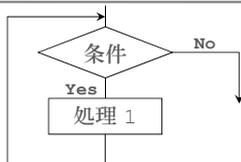
擬似言語とフローチャート表現(3)

➤ while 文 (前判定)

```

while (条件) {
  処理 1 ;
  処理 2 ;
  :
}

```

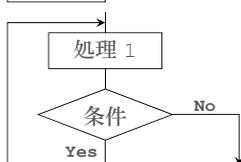


➤ do - while 文 (後判定)

```

do {
  処理 1 ;
  処理 2 ;
  :
} while (条件);

```



2008/10/3

第1講 アルゴリズムとは何か?

23

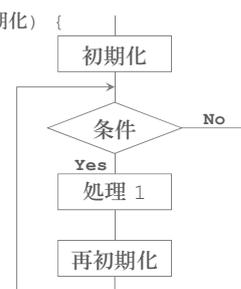
擬似言語とフローチャート表現(4)

➤ for 文

```

for (初期化 ; 条件 ; 再初期化) {
  処理 1 ;
  処理 2 ;
  :
}

```



2008/10/3

第1講 アルゴリズムとは何か?

24

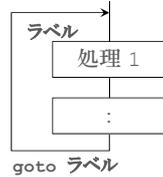
擬似言語とフローチャート表現(5)

▶ goto 文

```
goto ラベル ;
```

▶ 関数と return 文

```
関数名 (仮引数の並び) {
    処理 1 ;
    処理 2 ;
    :
    処理 3 ;
    return (式) ;
}
```



2008/10/3

第1講 アルゴリズムとは何か?

25

実際にアルゴリズムを使う練習

▶ ユークリッドの互除法

- ▶ 明示的に示された世界最古のアルゴリズム
- ▶ 紀元前 300 年ごろに発見
- ▶ 2 つの数の最大公約数を求める

▶ 単純に最大公約数を求めるためには...

- ▶ 因数分解できたら良い
 - ▶ 30 と 18 なら $30 = 2 \times 3 \times 5$, $18 = 2 \times 3 \times 3$ なので, 最大公約数は $2 \times 3 = 6$
 - ▶ 大きな数の因数分解は非常に難しく時間がかかる
 - ▶ 2 から順にその数を割ってみる方法とか

2008/10/3

第1講 アルゴリズムとは何か?

26

ユークリッドの互除法

▶ 2 つの正整数 $m, n (m > n)$ の最大公約数を求める

1. m を n で割って, 余りを r とする.
2. $r = 0$ であれば終了. このとき n が最大公約数.
3. $m \leftarrow n$ とする. 次に $n \leftarrow r$ として 1. に戻る.

▶ 試してみる

- ▶ $m = 30, n = 18 \rightarrow m \div n = 1$ 余り 12
- ▶ $m = 18, n = 12 \rightarrow m \div n = 1$ 余り 6
- ▶ $m = 12, n = 6 \rightarrow m \div n = 2$ 余りなし
- ▶ 6 が最大公約数!! 合ってる!!

2008/10/3

第1講 アルゴリズムとは何か?

27

ユークリッドの互除法の正当性(1/2)

▶ 正当性を調べる

- ▶ $m \div n = a$ 余り r とすると
 $m = a \times n + r$

- ▶ m と n の最大公約数を G とすると r も G で割り切れないと上の式は成り立たない
- ▶ つまり, G は r の約数でもある

- ▶ 次に, n と r の最大公約数 H とすると, 上の式の右辺を H で割ると整数になる
- ▶ つまり, m も H で割り切れるので, H は m の約数でもある

2008/10/3

第1講 アルゴリズムとは何か?

28

ユークリッドの互除法の正当性(2/2)

▶ G も H も m, n, r の約数

- ▶ G は m, n の最大公約数
- ▶ H は n, r の最大公約数

▶ この場合 G と H が同じでないと n の最大公約数ということに矛盾する

▶ つまり $G=H$ であり, H を求めると G を求めたことになる

2008/10/3

第1講 アルゴリズムとは何か?

29

互除法の決定性, 有限性, 停止性

▶ 決定性

- ▶ 入力 m, n が同じなら, 割り算結果は同じなので, 必ず同じ結果になる

▶ 有限性

- ▶ 割り算や代入は有限時間で必ず実行できる

▶ 停止性

- ▶ 余りは絶対に割る数より小さくなる
- ▶ そのため, 最悪でも n 回繰り返せば n は 1 になり, m を割り切るので必ず終了する

2008/10/3

第1講 アルゴリズムとは何か?

30

互除法の実行例

▶ 試してみよう

- ▶ $m = 105, n = 75$
 - ▶ $105 \div 75 = 1 \dots 30$
 - ▶ $75 \div 30 = 2 \dots 15$
 - ▶ $30 \div 15 = 2 \dots 0$
- ▶ $m = 288, n = 52$
 - ▶ $288 \div 52 = 5 \dots 28$
 - ▶ $52 \div 28 = 1 \dots 24$
 - ▶ $28 \div 24 = 1 \dots 4$
 - ▶ $24 \div 4 = 6 \dots 0$

2008/10/3

第1講 アルゴリズムとは何か?

31

アルゴリズムは道具

- ▶ 別に正当性などを自分で証明できなくても、有効なアルゴリズムはそのまま使えば良い
 - ▶ 狩りに出るときは、鉄砲の作り方を知らなくても、使い方だけ知っていれば良いのと同じ
- ▶ 利用できるアルゴリズムは怖がらず使っていこう

2008/10/3

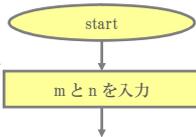
第1講 アルゴリズムとは何か?

32

フローチャートを描く練習

- ▶ フローチャートを使ってユークリッドの互除法を書いてみる
- ▶ ユークリッドの互除法

- ▶ 2つの正整数 $m, n (m > n)$ の最大公約数を求める
 1. m を n で割って、余りを r とする。
 2. $r = 0$ であれば終了。このとき n が最大公約数。
 3. $m \leftarrow n$ とする。次に $n \leftarrow r$ として 1. に戻る。
- ▶ まず start
- ▶ 入力は m と n



2008/10/3

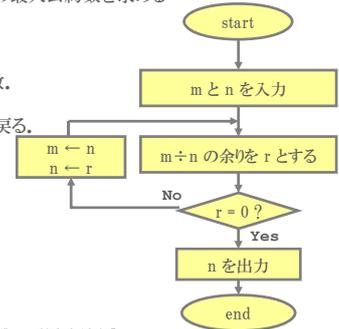
第1講 アルゴリズムとは何か?

33

互除法のフローチャート

- ▶ 2つの正整数 $m, n (m > n)$ の最大公約数を求める
 1. m を n で割って、余りを r とする。
 2. $r = 0$ であれば終了。このとき n が最大公約数。
 3. $m \leftarrow n$ とする。次に $n \leftarrow r$ として 1. に戻る。

- ▶ r を計算する
- ▶ 余りが 0 か調べる
- ▶ 0 でないなら n を m , r を n として繰り返す
- ▶ 0 なら n を出力
- ▶ 最後は end



2008/10/3

第1講 アルゴリズムとは何か?

34

擬似言語を書く練習

- ▶ 擬似言語を使ってユークリッドの互除法を書いてみる

▶ ユークリッドの互除法

- ▶ 2つの正整数 $m, n (m > n)$ の最大公約数を求める
 1. $r \leftarrow m \div n$ の余り
 2. $r = 0$ であれば n を出力して終了
 3. $m \leftarrow n, n \leftarrow r$ として 1. に戻る

- ▶ 関数に名前をつける
- ▶ 入力は引数として与える
- ▶ r を求める
- ▶ 変数の代入
- ▶ 終了条件の判定
- ▶ 出力して終わる

```

int gcd(int m, int n)
{
    do{
        r ← m ÷ n の余り;
        m ← n;
        n ← r;
    }while(r <> 0);
    return m;
}
    
```

2008/10/3

第1講 アルゴリズムとは何か?

35

擬似言語はいろいろな書き方がある

- ▶ r を求める部分を C 言語や Pascal のようにしても良い
 - ▶ $r \leftarrow m \% n;$
 - ▶ $r \leftarrow m \bmod n;$
- ▶ 同じ処理を行うなら異なる制御構造を使って良い

```

while(TRUE){
    r ← m % n;
    if(r = 0)
        goto label_1;
    m ← n;
    n ← r;
}
label_1:
    
```

```

do{
    r ← m % n;
    m ← n;
    n ← r;
}while(r <> 0);
    
```

```

r ← 1;
while(rが0でない){
    r ← m ÷ n の余り;
    m ← n;
    n ← r;
}
    
```

```

for(r ← m % n; r <> 0; m ← n, n ← r){
    r ← m % n;
}
    
```

- ▶ できるだけ読みやすくするのも大事

2008/10/3

第1講 アルゴリズムとは何か?

36

第 1 講のまとめ

- ▶ アルゴリズムの定義
 - ▶ 入力と出力
 - ▶ 正当性, 決定性, 有限性, 停止性
- ▶ ユークリッドの互除法
- ▶ フローチャートの描き方
- ▶ 擬似言語の書き方

2008/10/3

第1講 アルゴリズムとは何か?

37