

情報処理技法 (Javaプログラミング)2

第3回 アルゴリズム(1)

人間科学科コミュニケーション専攻 白銀 純子

Copyright (C) Junko Shirouane, Tokyo Woman's Christian University, 2016 All rights reserved.

第3回の内容

▼アルゴリズム

Copyright (C) Junko Shirouane, Tokyo Woman's Christian University, 2016 All rights reserved.

前回の出席課題の解答

▼下記のプログラムを実行したとき、確実に標準出力に出力されるメッセージは何かを答えなさい。

```

String line[] = new String[5];
int k;
FileReader fr = new FileReader("sample.txt");
BufferedReader br = new BufferedReader(fr);
for(i = 0; i < 100; i = i + 1) {
    line[i] = br.readLine();
}
br.close();
br.close();
catch (IOException e) {
    System.out.println("指定されたファイルが手続が完了していません。");
}
catch (NumberFormatException e) {
    System.out.println("数値を表す文字列が数値でない文字列を数値に変換しようとしています。");
}
catch (FileNotFoundException e) {
    System.out.println("指定されたファイルが存在しません。");
}
catch (IOException e) {
    System.out.println("指定されたファイル読み込みことができません。");
}

```

Copyright (C) Junko Shirouane, Tokyo Woman's Christian University, 2016 All rights reserved.

アルゴリズム

Copyright (C) Junko Shirouane, Tokyo Woman's Christian University, 2016 All rights reserved.

アルゴリズムとは

▼アルゴリズム: ある問題を解決するときに必要な処理手順
▼プログラムを書くときには、必ずアルゴリズムを考える必要
▼プログラム: アルゴリズムをプログラミング言語を使って記述したもの

Ex. 焼きそば作りのアルゴリズム:

1. キャベツや肉などの具をフライパンで炒めなさい。
2. フライパンからいったん具を取り出しなさい。
3. めんをフライパンで炒めなさい。
4. 具をめんの入っているフライパンに戻しなさい。
5. 焼きそばソースを加えてさらに炒めなさい。

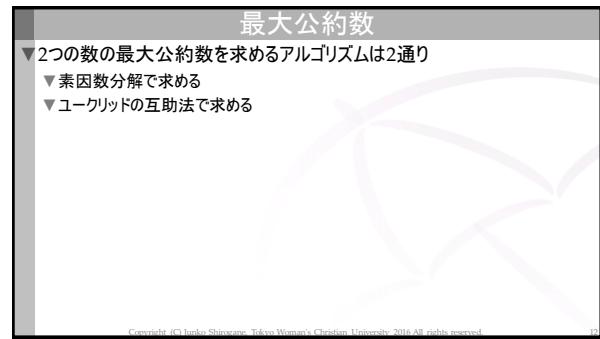
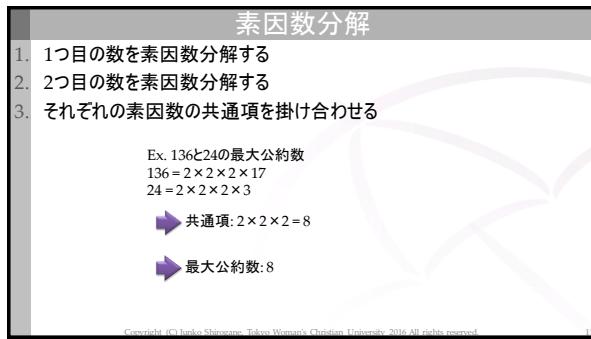
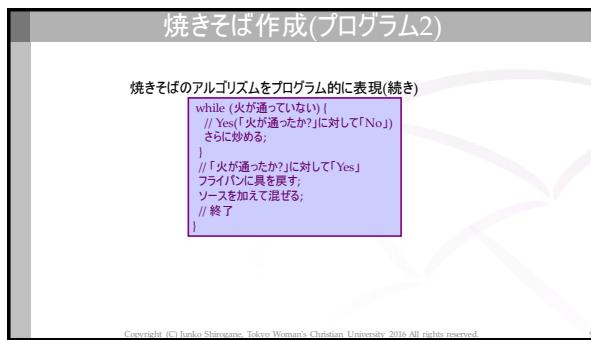
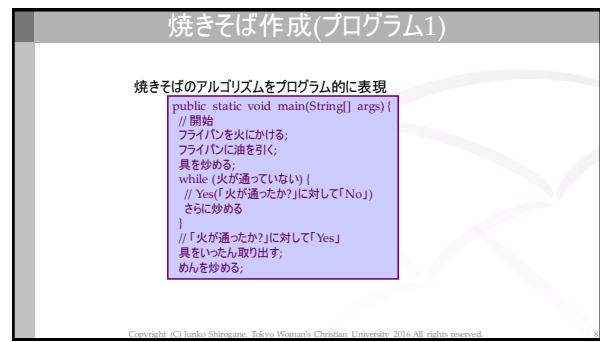
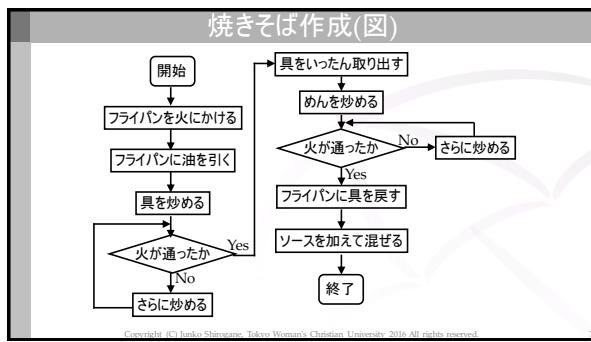
Copyright (C) Junko Shirouane, Tokyo Woman's Christian University, 2016 All rights reserved.

アルゴリズムの表現方法

▼文章で書く
▼箇条書きで書くことが多い
▼図で書く
▼プログラムで書く
▼プログラムが最終段階

▼プログラムで行う処理手順がわからなければ、まず文章で書き出して、それを詳細にしていく

Copyright (C) Junko Shirouane, Tokyo Woman's Christian University, 2016 All rights reserved.



ユークリッドの互除法

1. 2つの数のうち、大きい方を小さい方で割る
2. 1. の余りが0でない場合、その余りと小さい方の数を改めて2つの数とし、1. へ戻る
3. 1. の余りが0の場合、割った数が最大公約数となる

Ex. 136と24の最大公約数

$$\begin{array}{r} 136 \div 24 = 5 \dots 16 \\ \downarrow \\ 24 \div 16 = 1 \dots 8 \\ \downarrow \\ 16 \div 8 = 2 \dots 0 \end{array}$$

最大公約数: 8

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

筆算のアルゴリズム

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

自然数の加算(1)

▼ 2つの自然数の和を求める = 筆算

▼ n 桁目の数の和を求め、10以上になった場合には $n+1$ 桁目に1を加える
→ 1から自然数の桁数まで繰り返す

▼ 準備

▼ 2つの自然数は同じ桁数とする

▼ 50と200の場合、050と200として考える

▼ 加えた結果を入れる変数(配列が便利)を用意する

▼ 繰り上がりを入れる変数を用意し、0を代入しておく

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

自然数の加算(2)

1. 1の位から最上位の位まで、a. ~d. を繰り返す

- その位の2つの数の和を求める
- 下位からの繰り上がりがあればそれも加える
- その和の1桁目を、求める和のその位の結果とする
- その和の2桁目を、次の桁への繰り上がりとする

2. 最上位からの繰り上がりがあった場合は、求める和のその次(上位)の位の数とする

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

ATMのアルゴリズム

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

ATM(1)

▼ 人間から見える手順

1. 「引き出し」ボタンを押す
2. キャッシュカードを入れる
3. 暗証番号を入力する
4. 引き出し金額を入力する
5. 現金を取り出す

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

ATM(2)

▼ ATM側での処理手順

1. 初期画面を表示してボタンが押されるのを待つ
2. 押されたボタンによってサービスの種類を判別する
 - a. 「引き出し」ボタンの場合、3.へ進む
 - b. 「預け入れ」ボタンの場合、4.へ進む
3. サービスの種類が「引き出し」の処理
4. サービスの種類が「預け入れ」の処理
-
5. 「ありがとうございました」の画面を表示して、1.へ戻る

Copyright (C) Junko Shirozane, Tokyo Women's Christian University, 2016 All rights reserved.

ATM(3)

▼ 「引き出し」の処理手順

- a. カードを入れてくださいと表示する
- b. 適切なカードかどうかを調べ、キャッシュカードではないカードであれば一.へ進む
 - 一. 入れられたカードを排出し、「キャッシュカードを入れなおしてください。」と表示する
 - 二. b.へ戻る
- c. 暗証番号を調べ、番号が間違っていたら一.へ進む
 - 一. 「暗証番号を入力しなおしてください。」と表示する
 - 二. c.へ戻る
- d. 「引き出す金額を入力してください。」と表示し、引き出し金額を入力してもらう
- e. 引き出し金額を調べる

Copyright (C) Junko Shirozane, Tokyo Women's Christian University, 2016 All rights reserved.

ATM(3)

▼ 「引き出し」の処理手順の続き

- f. 引き出し金額が残高よりも多ければ、一.へ進む
 - 一. 「残高を超えています。引き出し金額を入力しなおしてください。」と表示する
 - 二. f.へ戻る
- g. 出金処理を行い、預金残高を更新する
- h. 預金残高を表示し、5.へ進む

Copyright (C) Junko Shirozane, Tokyo Women's Christian University, 2016 All rights reserved.

有名なアルゴリズム

Copyright (C) Junko Shirozane, Tokyo Women's Christian University, 2016 All rights reserved.

並べ替え(ソート)

▼ ソート: 複数の数を小さい or 大きい順に並べること

- ▼ バブルソート
- ▼ 選択ソート
- ▼ 併合ソート
- ▼ クイックソート
- ▼ etc.

Copyright (C) Junko Shirozane, Tokyo Women's Christian University, 2016 All rights reserved.

バブルソート(1)

▼ 前から2つずつ、数の大きさを比較して、小さい数を後ろに送っていく

▼ 最後まで調べると、最も小さな数が一番後ろにある

▼ この作業を、並べ替える数の個数だけ繰り返すと、数が大きい順に並ぶ

▼ $x_1 < x_2$ ならば、 x_1 と x_2 を入れ替える

▼ $x_2 < x_3$ ならば、 x_2 と x_3 を入れ替える

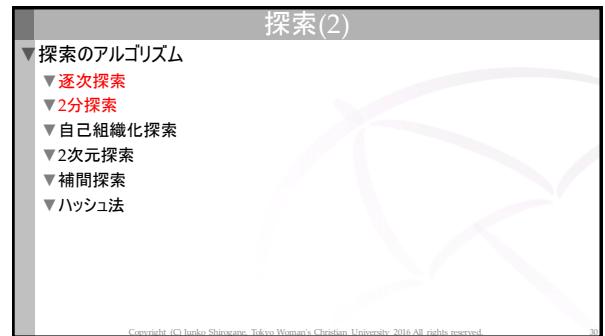
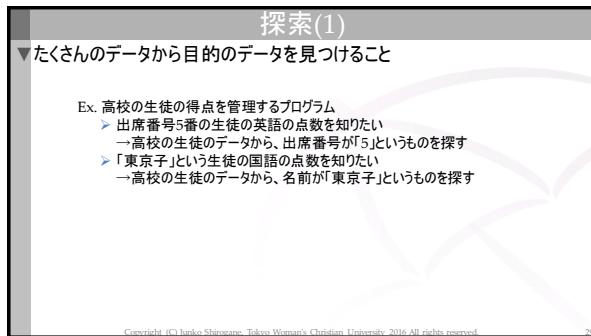
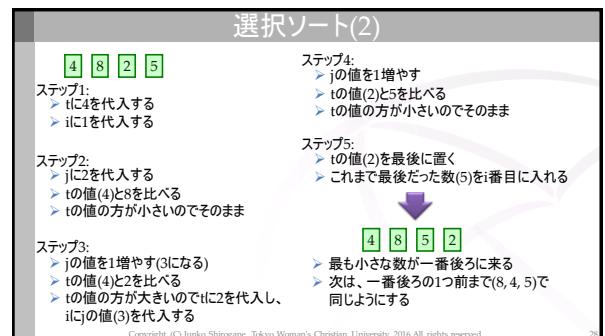
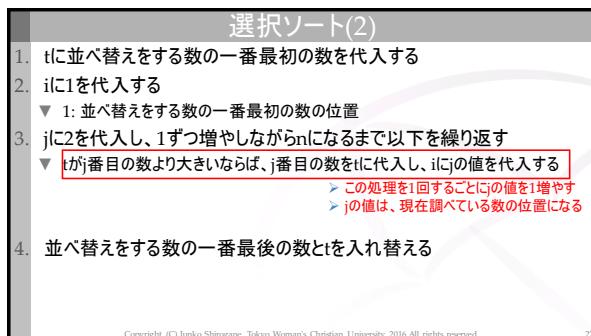
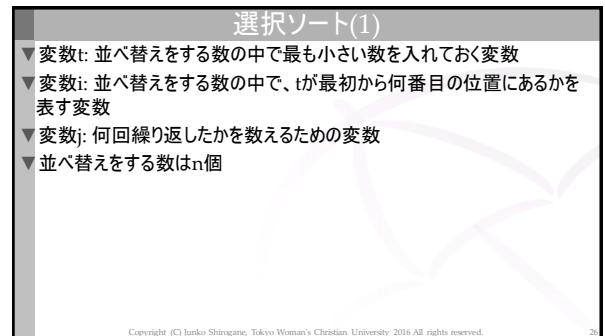
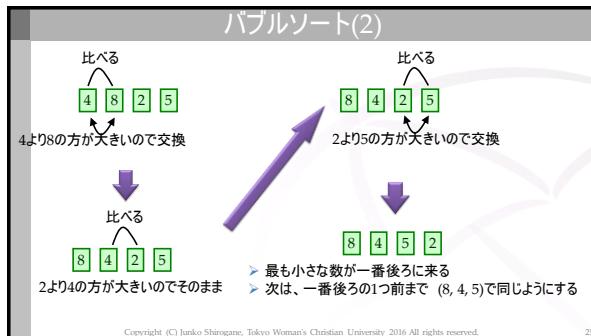
...

▼ $x_{n-1} < x_n$ ならば、 x_n と x_{n-1} を入れ替える

➤ n個の数でこの処理が終わった後にn番目に最も小さな数

➤ n-1回この処理を繰り返すことで、大きい順に数を並べ替え

Copyright (C) Junko Shirozane, Tokyo Women's Christian University, 2016 All rights reserved.



逐次探索

▼ データを前から順番に比較して探していく方法

1. データが配列a[0]～a[N]に入っている

2. 「a[i] = x」(xは探したいデータ)となる「i」を探す

▼ 「i」は0～Nの整数

▼ 欠点: データが配列の後ろのほうにある場合に見つけるまでに時間がかかる

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

2分探索(1)

▼ ソートされたデータの中から目的のデータを探す方法

1. データが配列a[0]～a[N]に入っている

2. 中央のデータ「a[(0+N)/2]」と「x」を比較する

▼ x: 探したいデータ

3. 2.の結果、a[m] < xの場合、データは右半分にx <= a[m]の場合、データは左半分に存在

▼ m: (0+N)/2

▼ 右半分: a[m+1]～a[N]

▼ 左半分: a[0]～a[m]

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

2分探索(2)

4. データが右半分にある場合、a[m+1]～a[N]で2.をする

5. データが左半分にある場合、a[0]～a[m]で2.をする

6. 「a[left]～a[right]」での探索で、「left = right」となると探索終了

7. 目的のデータは「a[left]」

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

2分探索(3)

「1, 2, 3, 4, 5, 6, 7, 8, 9, 10」の中から「9」を見つけよう

▶ a[(0+9)/2](値: 5)と「9」の方が大きいので、「9」は右半分(a[(0+9)/2+1]～a[9])の範囲にある

▶ a[(5+9)/2](値: 8)と「9」を比較し、「9」の方が大きいので、「9」は右半分(a[(5+9)/2+1]～a[9])の範囲にある

▶ a[(8+9)/2](値: 9)と「9」を比較し、「9」と同じなので、「9」は左半分(a[(5+9)/2]～a[(8+9)/2])の範囲にある

▶ a[(7+8)/2](値: 8)と「9」を比較し、「9」の方が大きいので、「9」は右半分(a[(7+8)/2+1]～a[8])の範囲にある

▶ 探索する配列の添え字が同じになったので、探索終了

▶ 結果: a[8]が回答

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

やってみよう!

逐次探索

1. 配ったカードをバラバラの順序にする

2. その中から「18」を探してみる

2分探索

1. 配ったカードを昇順に並べる

2. カードの中から「18」を探してみる

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

やってみよう!(1)

ユーリッドの互乗法のプログラム

▼ 2つの数を標準入力で

ATMの「引き出し」と「預け入れ」の機能のプログラム

▼ 預金残高と暗証番号は、あらかじめプログラム中に書いておく

▼ カードに関する処理はなしでOK

▼ 出金処理と入金処理は、標準入力で処理

▼ 出金: 「xx円出金し、yy円残っています。」と表示

▼ 入金: 入金する金額を標準入力で入力

Copyright (C) Junko Shirozane, Tokyo Woman's Christian University, 2016 All rights reserved.

やってみよう!(2)

▼「アルゴロジック」という教材に挑戦してみよう

▼<http://home.jeita.or.jp/is/highschool/algo/index.html>

▼アルゴリズムの考え方を身につけるのに良い教材

Copyright (C) Junko Shirouze, Tokyo Woman's Christian University, 2016 All rights reserved.