

# アルゴリズムとデータ構造

第1回 アルゴリズムとデータ構造概論

# 講義の概要

2

## □ 概要

- プログラミングで問題解決するための「考え方」を学ぶ
- プログラミング言語は、Javaを使用し、新しい文法も扱う

## □ 成績評価

- 中間・期末テスト 60%
- 課題提出 30%
- 授業中課題 10%

## □ 難易度に注意

- 目安は基本情報技術者の科目B試験まで合格するレベル
- 考える力を身に付ける科目なので、**答えは簡単に教えません**
- 「プロI・II」がB評価以下の人には、**課題に膨大な時間が必要**

# 想定する履修者

3

- 履修を勧める学生
  - 卒研や大学院の研究で、大きなプログラムを作つてみたい人
  - 将来IT系の会社などに入り、プロとして開発の仕事をしたい人
- 履修を勧めない学生
  - 自分で考えるよりも、教えてもらって覚える勉強がしたい人…
  - 資格の取得が目標で、合格に必要な知識だけを学びたい人…
  - 卒業できればよく、就職先や職種には特にこだわりがない人…
  - 第1回の演習課題をやってみて、前提知識が十分でない人…
- 授業内容について
  - 情報系の大学卒業者には“常識”とされる知識とスキルを学ぶ
  - プログラミングの経験がまだ少ない人は、来年の履修を推奨

# 生成AI等の使用制限

4

## □ 生成AI等の使用

- この科目は、アルゴリズムの基礎・基本形を学ぶので、生成AI (ChatGPT等)によるコード(プログラム)の生成は一切禁止
- 他人(ネット等)のコードの一部をコピーすることも禁止
- 開発環境のアシスト機能による構文生成も使用しないで下さい
- (英作文の練習でAIに翻訳させたら意味がないと同様)

## □ 許容される使い方

- メソッド(`println`等)の使い方やスペルを教えてもらう
- コード生成ではなく、日本語でアルゴリズムを教えてもらう
- 質問の例:「配列の内容を逆順に並べ替えるアルゴリズムを、プログラミング言語ではなく日本語で教えてください」
- 教科書やネットの資料をよく読んで理解し、参考にして解答する
- 上記のような使い方でも、必ずスクショや写真を添付すること

# 第1回のキーワード

5

## アルゴリズム関係

- アルゴリズム(algorithm)
- データ構造  
(data structure)
- 確率(probability)
- 期待値(expected value)
- 値の交換
- 再帰(recursion)
- ビット演算  
(bitwise operation)

## Java関係

- 配列

# アルゴリズム

6

## □ アルゴリズムとは？

- 同種の問題を解決するための計算の手順
- 一般的には、条件分岐や繰り返しを含み、  
1本の「公式」では表せないもの ⇒ プログラムで表現する
- 対象となるデータの構造(格納形式)と密接な関係がある

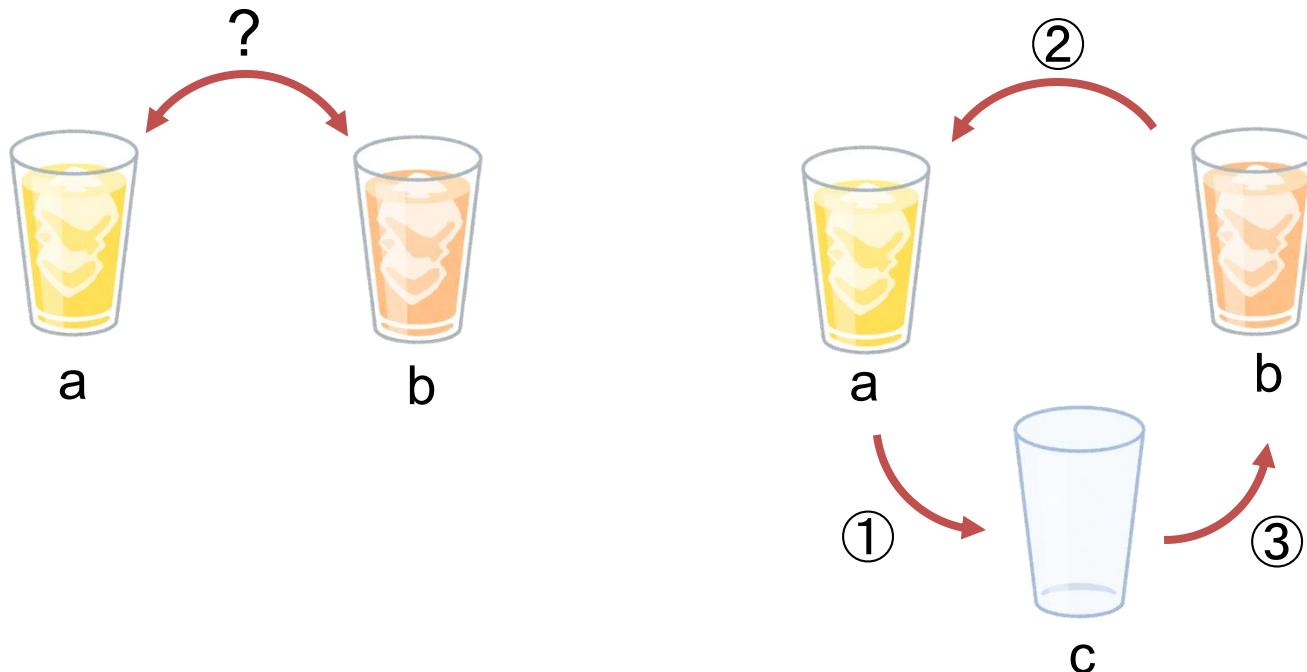
## □ 簡単なアルゴリズムの例

- 2つの変数の内容を交換する
- 分母が異なる分数を通分して足し算する
- 10進数を2進数に変換する
- 数値の配列の中から、最大値を選び出す

# 値の交換

7

- 2つの変数の値を交換するには
  - 内容を破壊しない手順が必要



# アルゴリズムとデータ構造

8

- よいアルゴリズムとは?
  - 速い(実行時間) ⇒ 時間計算量が少ない
  - 小さい(使用メモリ) ⇒ 空間計算量が少ない
  - これらを両立させるのが理想的だが、難しいことが多い
- データ構造
  - データを格納する構造(データの並べ方、つなぎ方など)を工夫すると、計算処理を高速化・小容量化できる
  - 単純なデータ構造: 配列
  - 動的なデータ構造: リスト構造、木構造、グラフ構造
  - 本科目の後半のテーマ

# 確率

9

- 確率(probability)
  - 確率 = 何かの事象が起きる“確かさ”(蓋然性)の度合い
  - 確率  $1/10$  = (同じ状況なら)平均して10回に1回起きる
  - 確率は割合の一種(起こり得る全ての事象の確率の合計は1)
- 「同様に確からしい」
  - 対称性によって、事象が起きる確かさが、理論的に等しいこと
  - 対称なサイコロなら、1~6の各目が出る確かさは等しい( $1/6$ )
- 大数の法則
  - 試行の回数を増やせば増やすほど、ある事象が起きる回数の割合(事象の回数 ÷ 総試行回数)は、その(真の)確率に近づく
  - 確率 = 無限回試行したときの、試行1回あたりに起きる回数

# 期待値

10

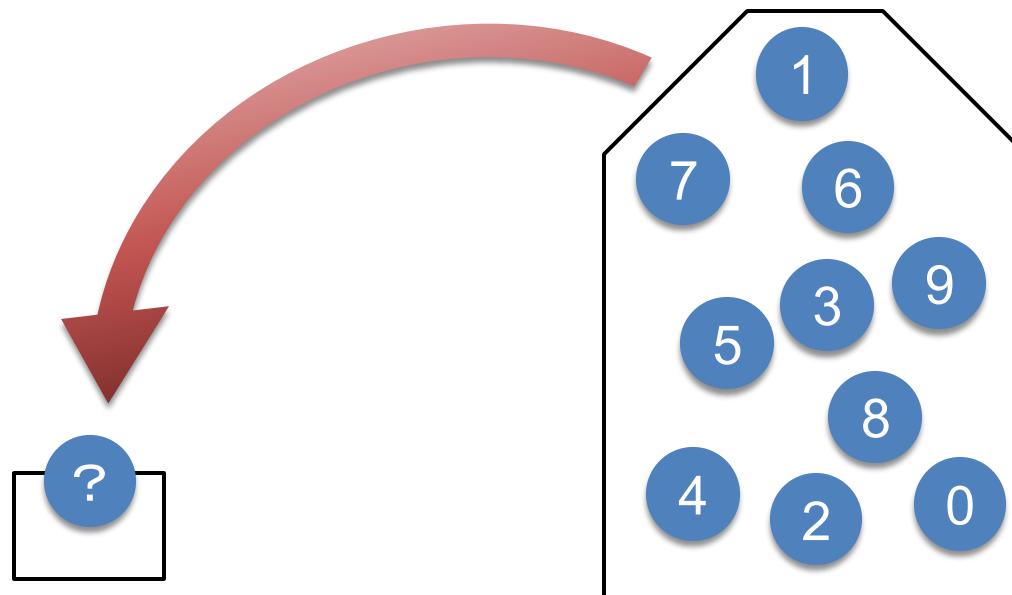
- 期待値
  - ギャンブルにたとえると、同じ状況の1回の賭け（試行）で儲けられる結果の平均値
  - {ある事象での値（儲け）×その事象の確率}の総和
- 考え方
  - 例えば、コインを投げて、もし表が出れば20ドルを得て、裏が出れば10ドルを失うとする
  - 1000回試行をする場合、表も裏も約500回ずつ出るので $20 \times 500 + (-10) \times 500 = 5,000$ ドルの収支が期待できる
  - 無限回の試行を考えると、確率（割合）を使って1回あたり $20 \times 0.5 + (-10) \times 0.5 = 5$ ドルと「期待値」が計算できる

# 確率と期待値の例

11

## □ 問題

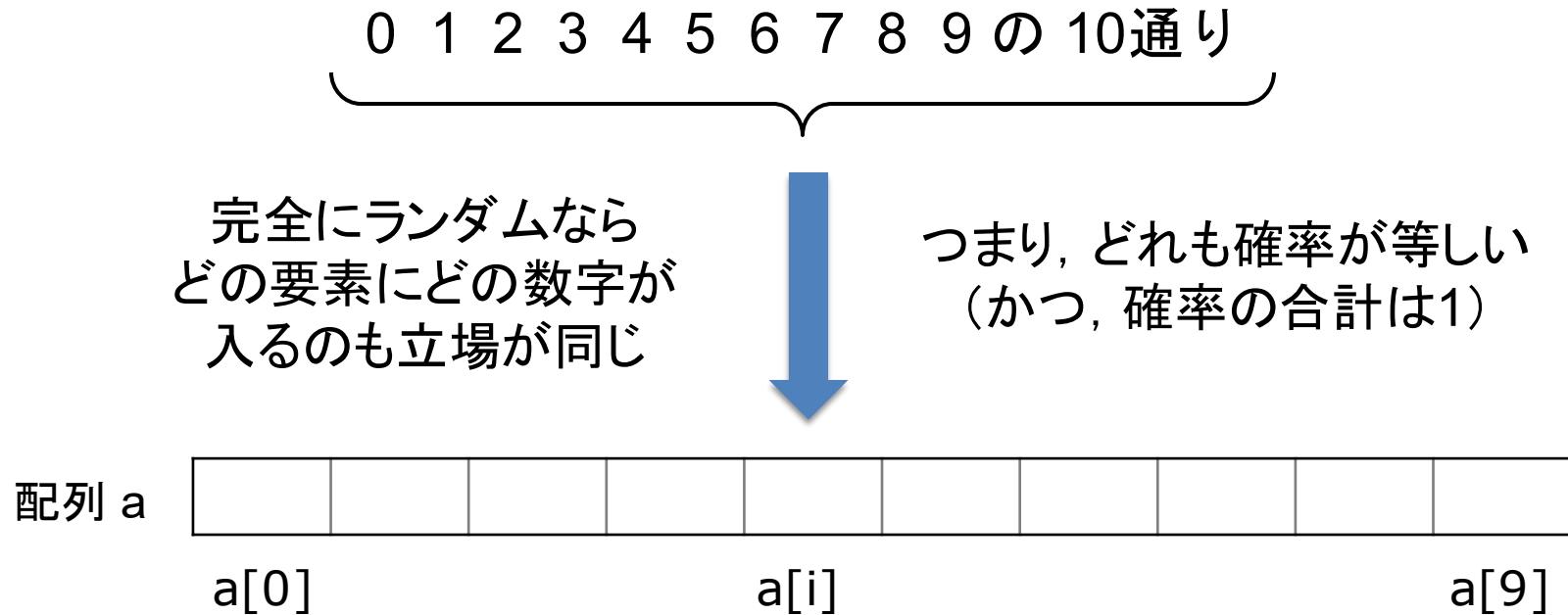
- 「0」～「9」の数字が書かれた玉が1個ずつ入った袋から、ランダムに玉を1個取り出し、箱に入れる
- 各玉が入る確率は？（「同様に確からしい」として）
- 数字を得点とみなすとその期待値は？



# 配列要素に関する確率

12

- 単純な確率は対称性で考える
  - 何と何が「同様に確からしい」(対称的な)関係か考察する



どの  $a[i]$  でも、0~9の特定の数字が入る確率は等しい  $\Rightarrow 1/10$

# 筆算のアルゴリズム

13

## □ 筆算と時間の計算は共通の方法

- 下の位(桁)から順に計算し、繰り上がり(繰り下がり)があれば、1つ上の位に加算(減算)していく
- 非常に桁数が多い整数を扱うプログラムにも応用される

$$\begin{array}{r} & 1 & 7 & 4 \\ + & & 5 & 9 \\ \hline & 2 & 3 & 3 \end{array}$$

↔

下の位から上の位へ計算

$$\begin{array}{r} 18\text{時間} & 34\text{分} & 48\text{秒} \\ +) 9\text{時間} & 53\text{分} & 35\text{秒} \\ \hline 1\text{日} & 4\text{時間} & 28\text{分} & 23\text{秒} \end{array}$$

↔

下の位から上の位へ計算

# 再帰

14

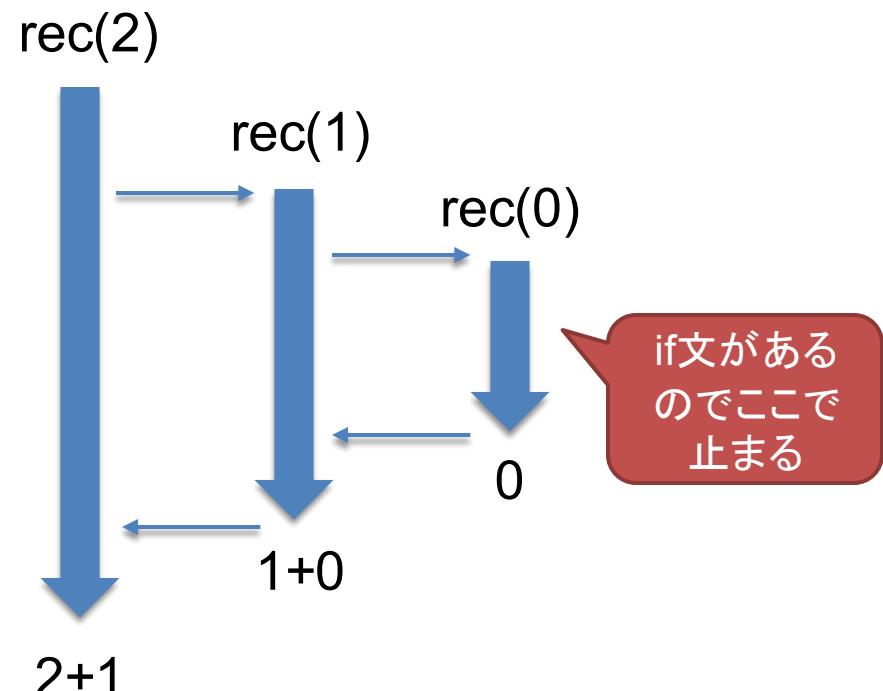
## □ 再帰とは

- 関数(メソッド)が、自分自身(の別のコピー)を呼ぶこと
- 実行過程では、それぞれの途中経過が保持されている
- 再帰処理の中では、再帰の終了判定が必要

```
int rec(int n) {
    System.out.println(n);

    // n=0ならもう再帰しない
    if (n == 0) return 0;

    return n + rec(n - 1);
}
```



# ビット演算

15

- コンピュータの内部表現は2進数
    - 例)1の内部表現

- 例) 1058の内部表現(1058=1024+32+2=2<sup>10</sup>+2<sup>5</sup>+2<sup>1</sup>)

MSB | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 | LSB

第10ビット      第5ビット      第1ビット

- ## □ ビット演算子

演算	説明	演算	説明
$x   y$	対応ビットごとに論理和(OR)	$\sim x$	ビットごとに論理否定(ビットの反転)
$x & y$	対応ビットごとに論理積(AND)	$x >> n$	全体を右にn桁ずらす(ビットシフト)
$x ^ y$	対応ビットごとに排他的論理和(XOR)	$x << n$	全体を左にn桁ずらす(ビットシフト)

# “参照型”の復習

16

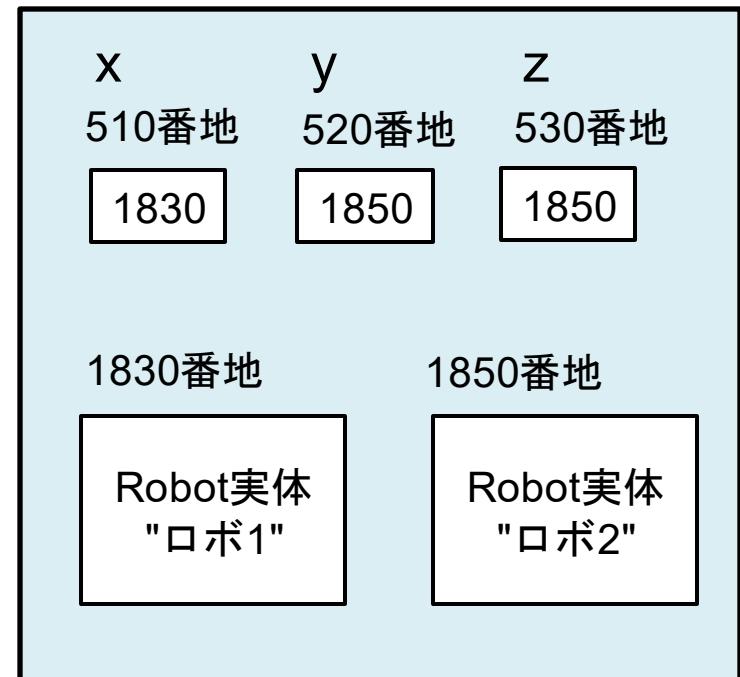
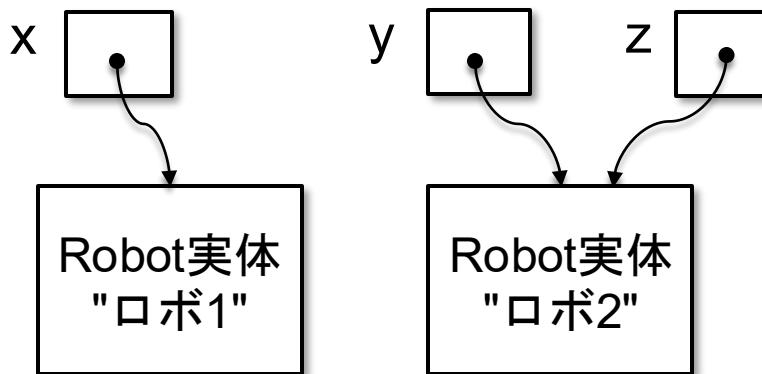
- Javaでは、クラス型や配列型の変数やフィールドは、インスタンス(実体)への参照(=つながり)しか持たない

- 変数とインスタンスの関係

```
Robot x = new Robot("ロボ1");
```

```
Robot y = new Robot("ロボ2");
```

```
Robot z = y;
```



メモリ内でのイメージ(番地は適当)