

第1回のキーワード

1

アルゴリズム関係

- アルゴリズム (algorithm)
- データ構造
(data structures)
- 確率 (probability)
- 期待値 (expected value)
- 再帰 (recursion)

Java関係

- 配列

アルゴリズム

2

- 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 ⇒ 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

アルゴリズムとデータ構造

3

□ 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
- これらは、両立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

4

- 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
- 「同様に確からしい」
 - ▣ 事象が起きる確かさが, (理論的に)等しいということ
 - ▣ 対称なコインなら, 表が出る確かさと裏が出る確かさは同じ
 - ▣ 対称なサイコロなら, 全ての目が出る事象は同様に確からしい
- 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 $\div$ 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

5

□ 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
- { 事象*i*での値(儲け) × 事象*i*の確率 } の総和

□ 考え方

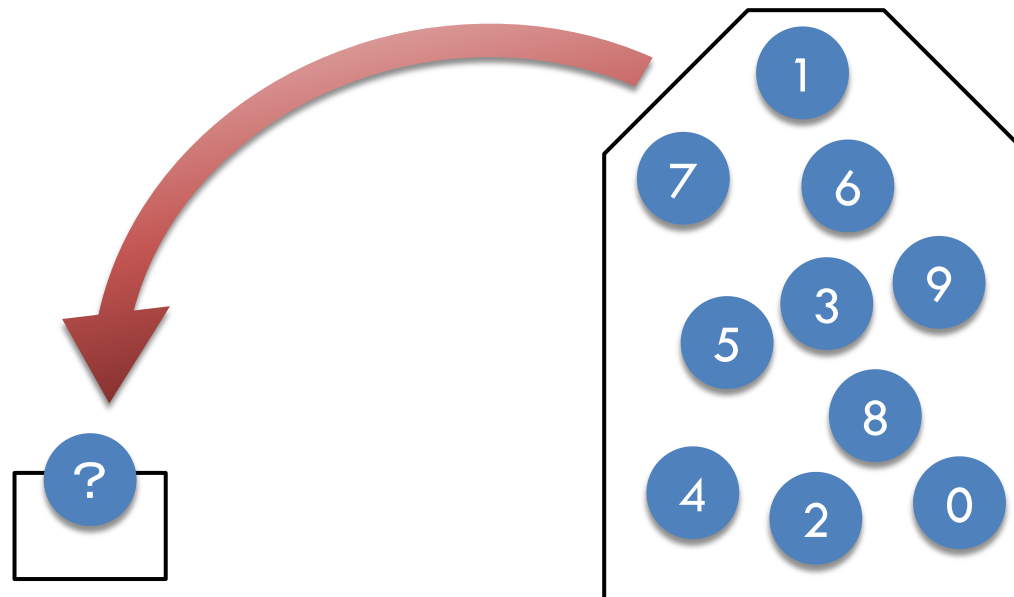
- 例えば, コインを投げて, もし表が出れば20ドルを得て, 裏が出れば10ドルを失うとする
- 1000回試行したとすると, 表も裏も約500回ずつ出るので $20 \times 500 + (-10) \times 500 = 5,000$ ドル の収支が期待できる
- 無限回試行したとして1回あたりで考えると, 確率を使って $20 \times 0.5 + (-10) \times 0.5 = 5$ ドル と期待値が計算できる

確率と期待値の例

6

□ 問題

- 「0」～「9」の玉が入った袋から、ランダムに1つ取り出し、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？



再帰

7

□ 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
- 必ず, 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 rec(n - 1) + n;  
}
```

