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

- 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
  - 中間テスト　　35%
  - 期末テスト　　35%
  - 課題提出 30%
- 難易度に注意
  - 基本情報技術者の科目B試験まで合格するレベルが目標
  - 思考力を習得する科目なので，答えは簡単に教えません
  - 「プロI・II」が評価B以下の人は，課題に膨大な時間が必要

第1回のキーワード

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

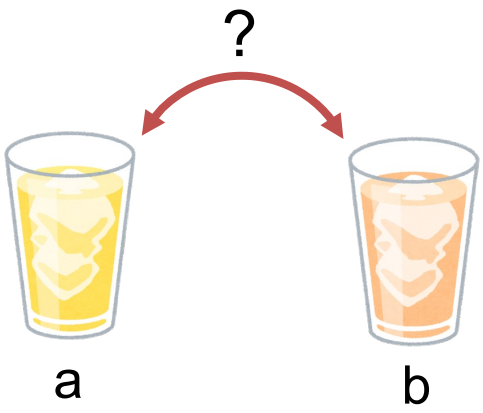
アルゴリズム

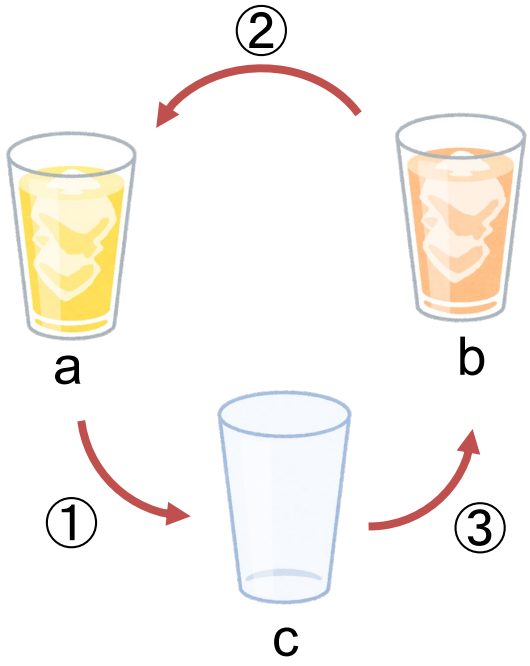
- アルゴリズムとは？
  - 同種の問題を解決するための計算の手順
  - 一般的には，条件分岐や繰り返しを含み，
    1本の「公式」では表せないもの ⇒ プログラムで表現する
  - 対象となるデータの構造（格納形式）と密接な関係がある

- 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

- 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

- 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

- 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（たいすう）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

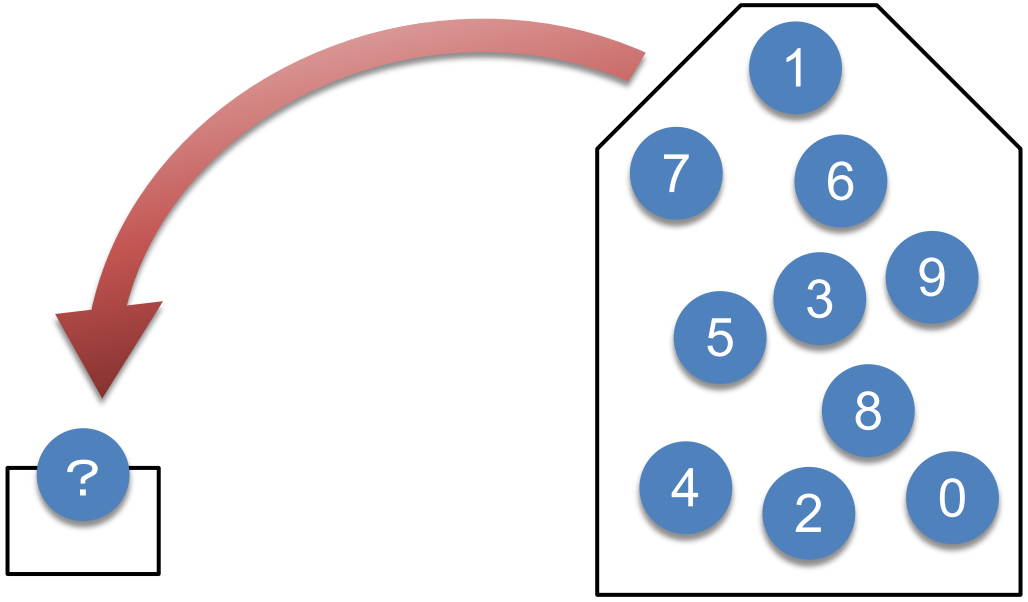
期待値

- 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 20×500+(-10)×500＝5,000ドル の収支が期待できる
  - 無限回試行の1回あたりを考えると，確率（割合）を使って 20×0.5＋(-10)×0.5＝5ドル と期待値が計算できる

確率と期待値の例

- 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？



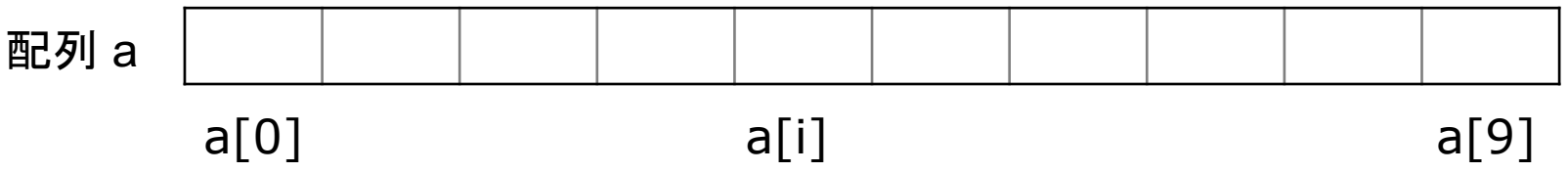
配列要素に関する確率

- 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

0 1 2 3 4 5 6 7 8 9 の 10通り

完全にランダムなら
どの要素にどの数字が
入るのも立場が同じ

つまり，どれも確率が等しい
（かつ，確率の合計は1）



どの a[i] でも，0〜9の特定の数字が入る確率は等しい ⇒ 1/10

筆算のアルゴリズム

- 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）があったら，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

	1	7	4
+		5	9
	2	3	3

下の位から上の位へ計算

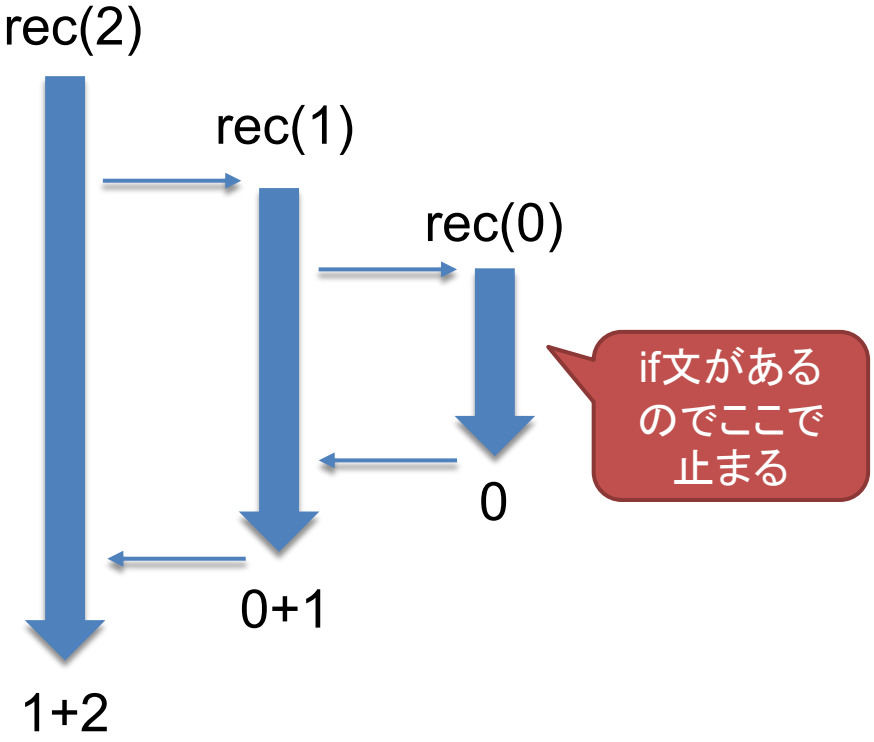
	18時間	34分	48秒
+)	9時間	53分	35秒
1日	4時間	28分	23秒

下の位から上の位へ計算

再帰

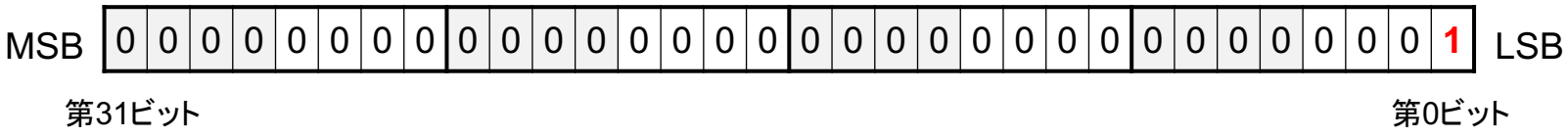
- 再帰とは
  - 関数（メソッド）が，自分自身（の別のコピー）を呼ぶこと
  - 再帰処理の中では，再帰の終了判定が必要

```
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 rec(n – 1) + n;
}
```

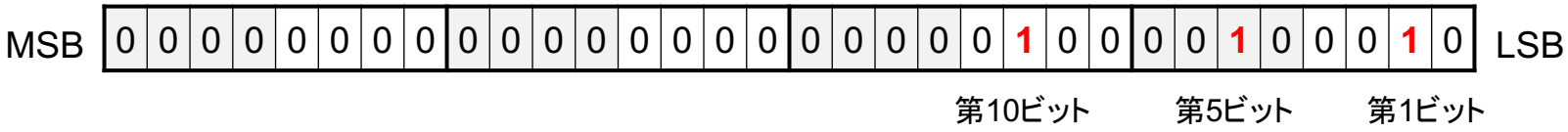


ビット演算

- 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=$2^{10}+2^5+2^1$）



- 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）	~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）