2章

アルゴリズムと計算量

M1 mokky

JIN NAKAZAWA LABORATORY

2.1 アルゴリズムとは

コンピュータ世界

データ処理
数値演算
組み合わせ計算
シミュレーション

日常世界

朝起きて
→着替えて
→朝食を食べて
→学校へ行く

問題に対する正しい出力を行い「停止する」、明確に定義された規則

2.2 問題とアルゴリズムの例

問題: Top 3

10人分のプレイヤーの得点が記録されたデータを読み込んで、その中から上位3人の得点を順に出力してください。ただし、得点は100点満点とします。

入力例

```
25  36  4  55  71  18  0  71  89  65
```

出力例

```
89  71  71
```

Algorithm1：3回探索する

Algorithm2：整列してから上位３つの得点を出力する

Algorithm3：各得点ごとの人数を数える

2.3 擬似コード

- 変数を英文字で表します。宣言文や型は省略します。
- 構造文として多くのプログラミング言語で使用できるif、while、for文を用います。
- ブロックは{ }ではなくインデント（字下げ）で表します。
- C/C++言語を模倣した演算子を用います。例えば、代入演算は=、等価演算は==、不等価演算子は!=で表します。論理演算子として、論理和||、論理積&&、否定!を用います。
- 配列Aの長さ（サイズ）は、A.lengthと表記します。
- 配列Aのi番目の要素をA[i]で表します。
- 配列の添え字は0オリジン[1]と1オリジンを場合によって使い分けます。

2.4 アルゴリズムの効率

高速に動作する優れたプログラムを書くには…

問題の性質を十分に考慮し、可能性のある全ての入力に対して効率良く動作する
アルゴリズムを考える必要がある

効率を評価する「ものさし」が必要

2.4.1 計算量の評価

- **時間計算量 (time complexity)**
  プログラムの実行に必要な時間を評価する
  計算機のプロセッサをどれだけ利用するかを見積もる

- **領域計算量 (space complexity)**
  プログラムの実行に必要な記憶領域を評価する
  計算機のメモリをどれだけ利用するかを見積もる

時間計算量と領域計算量のトレードオフやバランスを考えてアルゴリズムを設計

2.4.2 O表記法

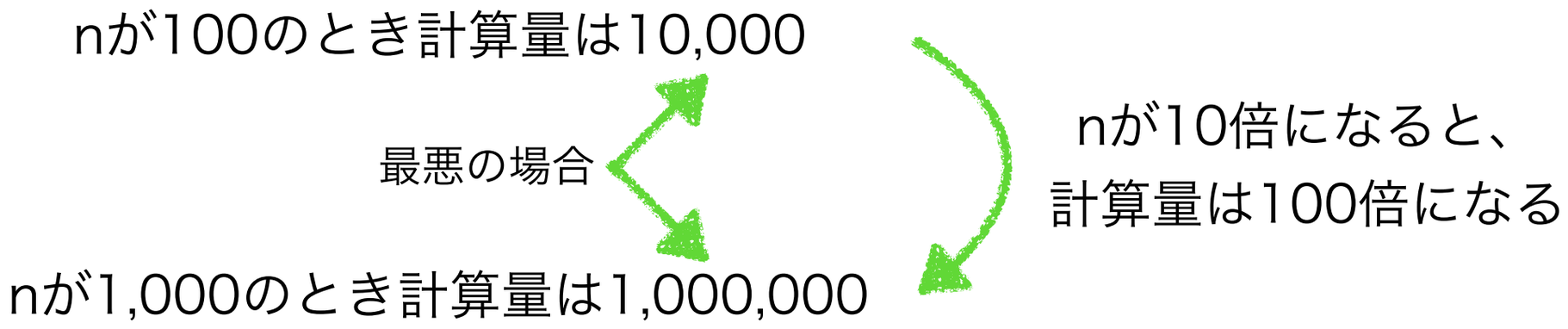
アルゴリズムの効率を評価する「ものさし」のひとつ

$O(g(n))$ → 計算量がg(n) に比例することを意味し、
「オーダーがg(n) である」と言う

2.4.2 O表記法

例 : n 件のデータを小さい順に整列してください。n は最大でも1,000件とします。

アルゴリズムAの計算量が　$O(n^2)$



2.4.2 O表記法

問題: Top N

m個の整数a_i (i = 1, 2, ..., m) が与えられます。その中で値が大きい順にn 個出力してください。

制約 $m \leq 1{,}000{,}000$
$n \leq 1{,}000$
$0 \leq a_i \leq 10^6$

Algorithm1 : $O(n \times m)$

Algorithm2 : $O(m \log m + n)$

Algorithm3 : $O(n + m + \max(a_i))$

各整数の値aiの最大値に比例するメモリ量

2.4.2 O表記法

アルゴリズムの計算量は、最善、平均、最悪の場合について見積もることができる

一般的なアルゴリズムの計算の設計においては最悪の計算量を見積もる

- 平均の計算量が最悪の計算量と同等になる場合が多い
- 最悪の計算量を見積もれば、それ以上の心配がいらない

2.4.3 計算量の比較

n	log$n$	$\sqrt{n}$	$n$log$n$	n^2	2^n	$n!$
5	2	2	10	25	32	120
10	3	3	30	100	1,024	3,628,800
20	4	4	80	400	1,048,576	約2.4×10^{18}
50	5	7	250	2,500	約10^{15}	約3.0×10^{64}
100	6	10	600	10,000	約10^{30}	約9.3×10^{157}
1,000	9	31	9,000	1,000,000	約10^{300}	約$4.0\times10^{2,567}$
10,000	13	100	130,000	100,000,000	約$10^{3,000}$	約$10^{35,660}$
100,000	16	316	1,600,000	10^{10}	約$10^{30,000}$	約$10^{456,574}$
1,000,000	19	1,000	19,000,000	10^{12}	約$10^{300,000}$	約$10^{5,565,709}$

よさげ　　きつい

2.4 導入問題

　FX取引では、異なる国の通貨を交換することで為替差の利益を得ることができます。例えば、1ドル100円の時に1,000ドル買い、価格変動により1ドル108円になった時に売ると、（108円－100円）×1,000ドル＝8,000円の利益を得ることができます。

　ある通貨について、時刻tにおける価格Rt (t = 0,1,2...,n - 1) が入力として与えられるので、
価格の差Rj－ Ri (ただし、j > i とする) の最大値を求めてください。

入力　最初の行に整数nが与えられます。続くn行に整数Rt (t = 0,1,2...,n - 1) が順番に与えられます。
出力　最大値を 1 行に出力してください。
制約　2 ≤ n ≤ 200,000、1 ≤ Rt ≤ 109

入力例 1

```
6
5
3
1
3
4
3
```

出力例 1

```
3
```

入力例２

```
3
4
3
2
```

出力例２

```
-1
```

2.4 導入問題

Program 2.1: 最大利益を求める素朴なアルゴリズム

```
1  for j が 1 から n-1 まで
2    for i が 0 から j-1 まで
3      maxv = (maxv と R[j]-R[i] のうち大きい方)
```

計算量は $O(n^2)$ **最大 4×10^{10}**

Program 2.2: 最大利益を求めるアルゴリズム

```
1  minv = R[0]
2  for j が 1 から n-1 まで
3    maxv = (maxv と R[j]-minv のうち大きい方)
4    minv = (minv と R[j] のうち小さい方)
```

計算量は $O(n)$ **最大 2×10^5**