

# [B7] ソーティング

大崎 博之

関西学院大学 理工学部 情報科学科

ohsaki@kwansei.ac.jp

## [B7] ソーティング

- ▶ 1. 単純選択法
- ▶ 2. 単純挿入法
- ▶ 3. クイックソート
- ▶ 4. 文字列や構造体のソート
- ▶ 5. バケツ法

これらを学ぶことに何の意味があるのか？

## ソート問題 (sorting problem)

- ▶ 入力:  $n$  個の値の順列  $(a_1, a_2, \dots, a_n)$
- ▶ 出力: 並べ替えられた値の順列  $(a'_1, a'_2, \dots, a'_n)$  ただし  $a'_1 \leq a'_2 \leq \dots \leq a'_n$

## 1. 単純選択法 (selection sort)

時間計算量:  $O(n^2)$

[http://en.wikipedia.org/wiki/Selection\\_sort](http://en.wikipedia.org/wiki/Selection_sort)

SELECTION-SORT (A)

```
1  for j    1 to length[A]-1
2      do min    j
3          for i    j+1 to length[A]
4              do if A[i] < A[min]
5                  then min    i
6          exchange A[i]    A[min]
```

1. 一番小さい値と  $A[1]$  を交換する
2.  $A[2..n]$  で一番小さい値と  $A[2]$  を交換する
3.  $A[3..n]$  で一番小さい値と  $A[3]$  を交換する

## 2. 単純挿入法 (insert sort)

時間計算量:  $O(n^2)$

[http://en.wikipedia.org/wiki/Insertion\\_sort](http://en.wikipedia.org/wiki/Insertion_sort)

```

INSERTION-SORT(A)
1  for j = 2 to length[A]
2      do key ← A[j]
3          # A[j] をソート済み順列 A[1 .. j-1] に挿入
4          i ← j-1
5          while i > 0 and A[i] > key
6              do A[i+1] ← A[i]
7                 i ← i-1
8          A[i+1] ← key
  
```

1.  $A[2]$  と  $A[1]$  を並び換える
2.  $A[3]$  を  $A[1..2]$  のどこかに挿入する
3.  $A[4]$  を  $A[1..3]$  のどこかに挿入する

### 3. クイックソート (quick sort)

時間計算量:  $O(n \log n)$

<http://en.wikipedia.org/wiki/Quicksort>

```

QUICKSORT(A, p, r)
1  if p < r
2      then q  PARTITION(A, p, r)
3          QUICKSORT(A, p, q-1)
4          QUICKSORT(A, q+1, r)

```

```

PARTITION(A, p, r)
1  x  A[r]
2  i  p-1
3  for j  p to r-1
4      do if A[j] <= x
5          then i  i+1
6              exchange A[i]
           A[j]
7  exchange A[i+1]  A[r]
8  return i+1

```

1.  $A[n]$  の値を基準  $x$  とする
2.  $x$  以下の値を  $A$  の前半に移動
3.  $x$  より大きい値を  $A$  の後半に移動
4. 前半と後半の境界 ( $x$  の位置) を  $q$  とする
5. 前半をクイックソート
6. 後半をクイックソート

## ソーティングを学ぶ意義

- ▶ プログラミングの本質は **アルゴリズム** と **データ構造**
  - ▶ プログラミングの成功/失敗は.....
    - ▶ 良い **アルゴリズム** を考えられるか?
    - ▶ 良い **データ構造** を考えられるか?
  - ▶ プログラミングテクニックは実は瑣末 (さまつ) な問題
  - ▶ だから大学の講義はとても重要!

ソーティング = 最も重要なアルゴリズムの一つ

設計した **アルゴリズム** を実現するための技術が必要

(実用上は、ソーティングのプログラムを自分で書く必要はない)