

アルゴリズム  
第13回授業  
“基本アルゴリズム整列”  
(教科書 Page 101-107)

山口雅樹 (CISSP)

<https://github.com/masakage/algorithm>

# 本日の進め方

- ・ 前回の復習（挿入法による整列）
- ・ 基本アルゴリズム（再帰処理）
- ・ 基本アルゴリズム（クイックソート整列）
- ・ まとめ

## 2-7 再帰

関数が “自分自身の関数” を呼び出すことを、再帰呼び出しという。  
 $N!$  ( $N$ の階乗) がよく知られている。

$N! = N \times (N-1) \times (N-2) \times (N-3) \times \dots \times 2 \times 1$  で表現される。  
(ただし、 $0! = 1$  と考える)

これは、 $N! =$   $N > 0$  のとき  $N \times (N-1)!$  ,  $N=0$  のとき  $1$  と考えられる

参考URL

<http://e-words.jp/w/%E5%86%8D%E5%B8%B0.html>

○プログラム名：再帰呼び出し /\* 教科書 102ページサンプル \*/

○整数型：Ret

●Ret ← Fact(5)

●表示処理(Ret)

○整数型：Fact(整数型：N)

▲N = 0

| ●return(1)

+ - - - - - - - - - -

| ●return(N) × Fact(N-1)

▼

5 x 4 x 3 x 2 x 1 = 120が結果となる。

5 x Fact (4)

4 x Fact(3)

3 x Fact(2)

2 x (Fact1)

1 x Fact(0)

## 2-8 クイックソート

クイックソートは、データの比較と交換回数が非常に少なく高速であるといわれています。ランダムに散らばっているデータに対して、最も効率良く並べ替えを実行します。

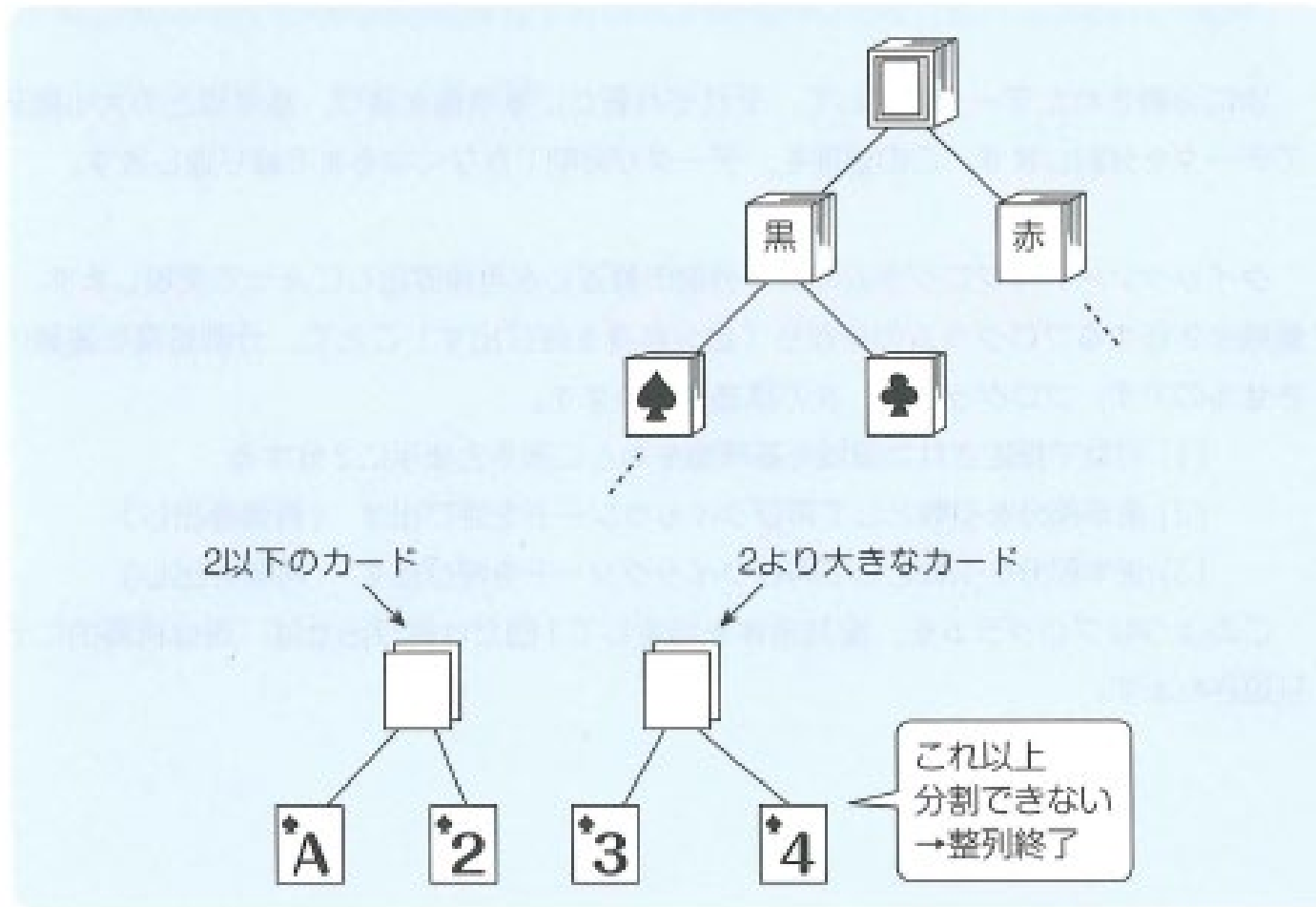
クイックソートは、高速であるとされている整列アルゴリズムで数多く採用されています。

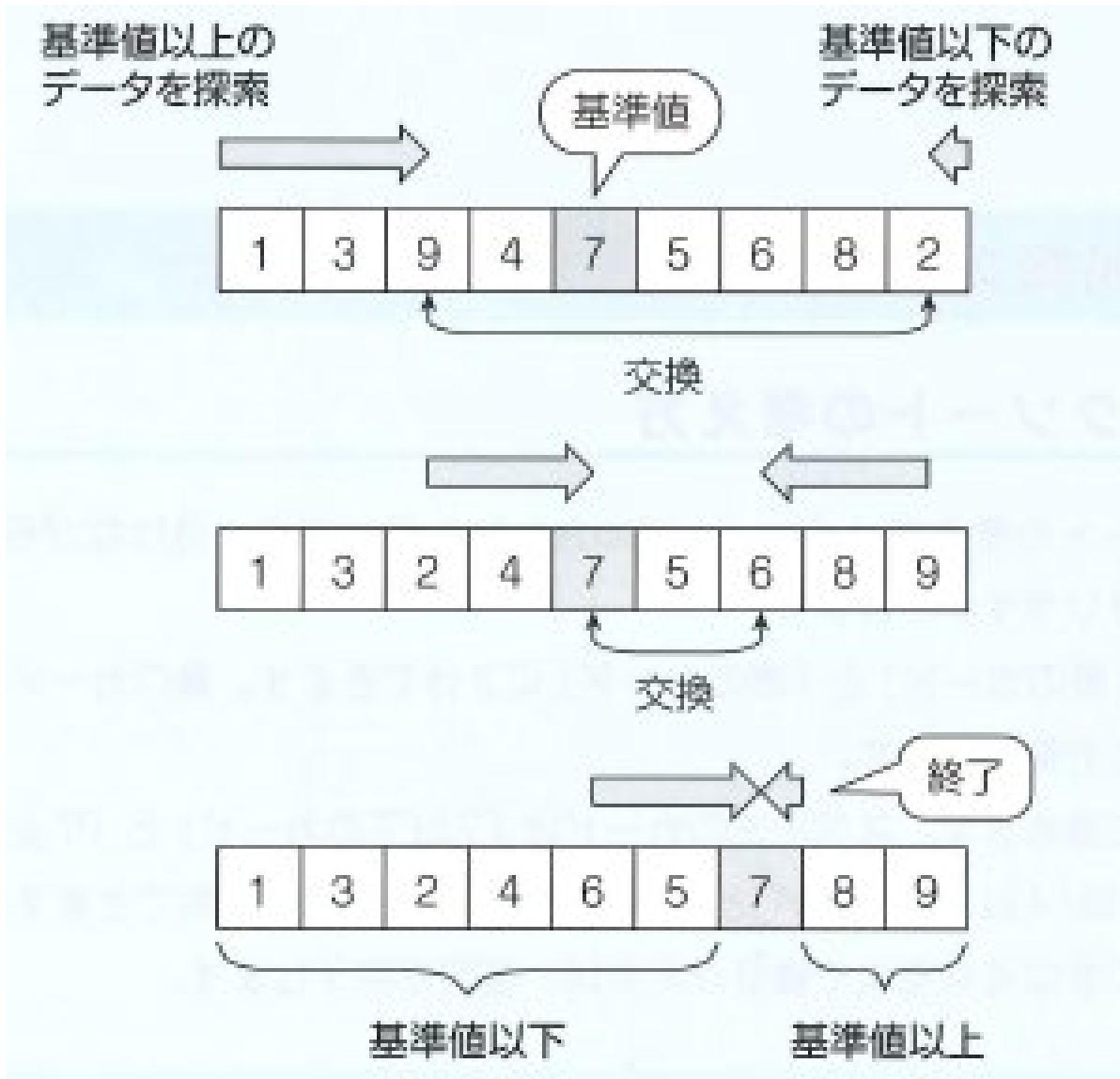
参考URL

<https://www.youtube.com/watch?v=a3yXjNhGwt0> (動画)

<http://www.ics.kagoshima-u.ac.jp/~fuchida/edu/algorithm/sort-algorithm/quick-sort.html> (鹿児島大学)

# クイックソートの考え方





データの中から基準値を選ぶ。

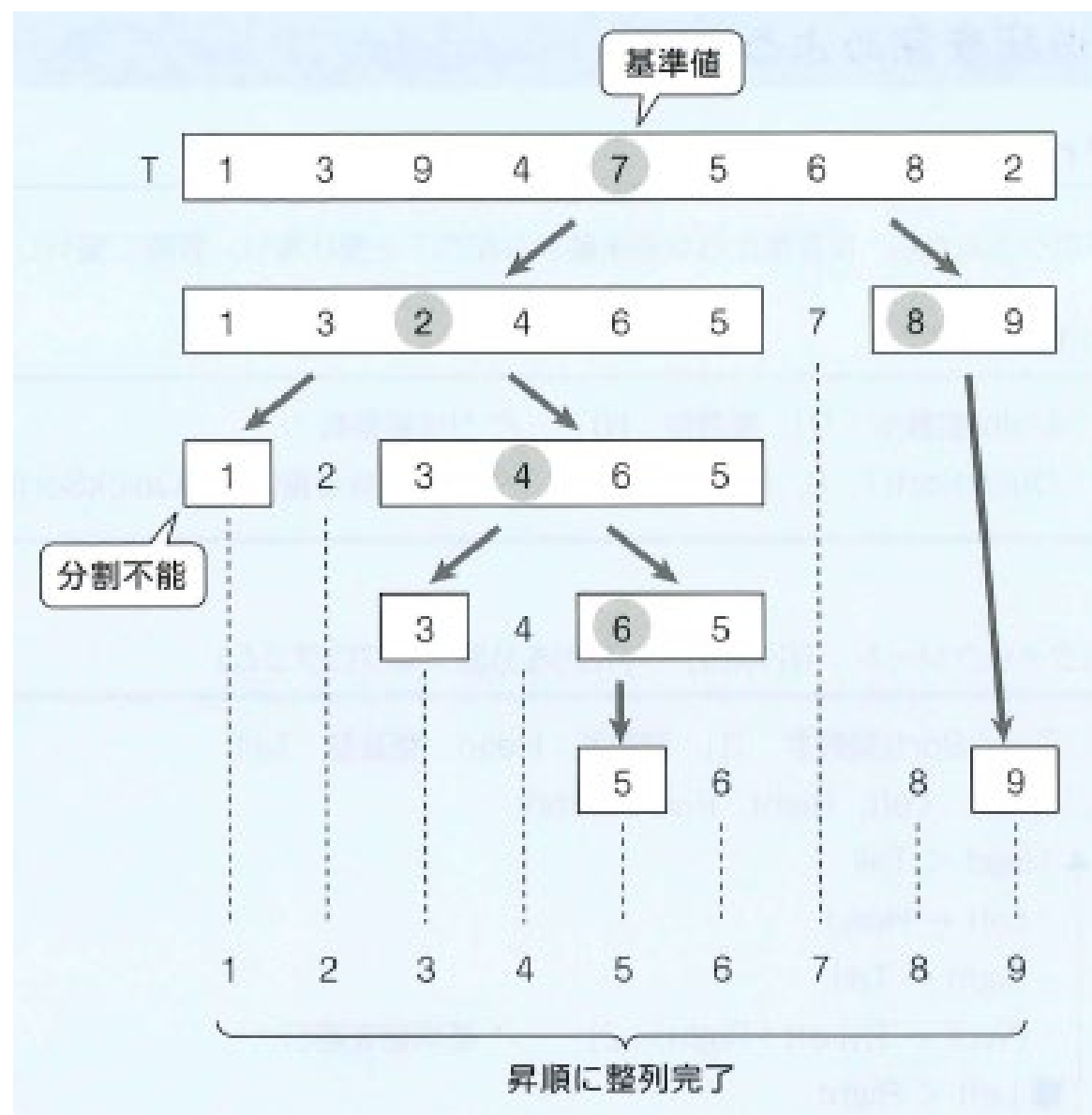
(教科書では、中央の値)

基準値を選んだら、基準値以下のデータは左側  
基準値以上のデータは、右側に配置する。

終われば、再度 基準値を計算して、  
再帰呼び出ししていく。

お手元のカードで試してください。





○QuickSort(整数型: T[], 整数型: Head, 整数型: Tail)

○整数型: Left, Right, Pivot, Work

▲ Head < Tail

・ Left ← Head

・ Right ← Tail

・ Pivot ←  $T[(\text{Left} + \text{Right}) \div 2]$  /\* 基準値を選ぶ \*/

■ Left < Right

■  $T[\text{Left}] < \text{Pivot}$  /\* 左から基準値以上の要素を探索 \*/

・ Left ← Left + 1

■

■  $T[\text{Right}] > \text{Pivot}$  /\* 右から基準値以下の要素を探索 \*/

・ Right ← Right - 1

■

▲ Left < Right /\* 左右の位置関係が正しければ交換 \*/

・ Work ←  $T[\text{Left}]$

・  $T[\text{Left}] \leftarrow T[\text{Right}]$

・  $T[\text{Right}] \leftarrow \text{Work}$

▼

■

・ QuickSort(T, Head, Left - 1) /\* 前半部分を指定した再帰呼出し \*/

・ QuickSort(T, Right + 1, Tail) /\* 後半部分を指定した再帰呼出し \*/

(主プログラム)

○Main(整数型: T[], 整数型: N) /\* Nは要素数 \*/

・ QuickSort(T, 0, N - 1) /\* 配列全体を指定してQuickSort呼出し \*/