第5回のキーワード

アルゴリズム関係

- クイックソート（quicksort）
- ピボット（軸）（pivot）
- 分割統治（divide and conquer）
- マージソート（merge sort）
- $O(n \log n)$

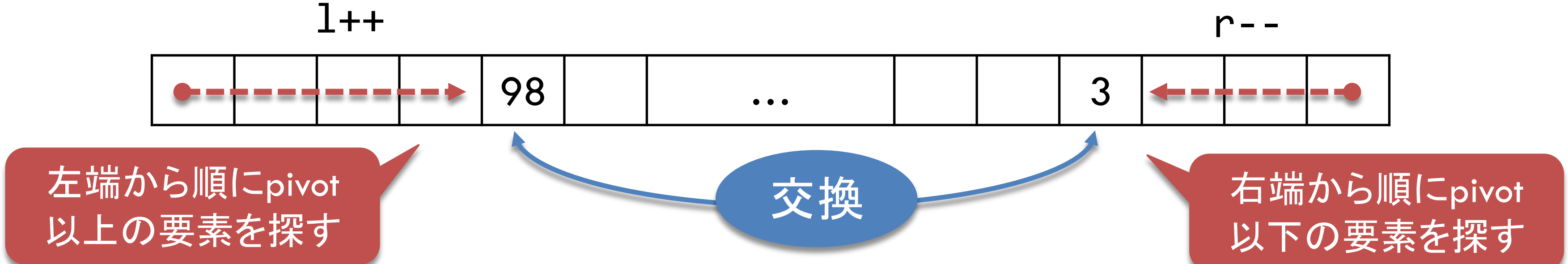
Java関係

- 無限ループ
  - while(true)
  - for(;;)

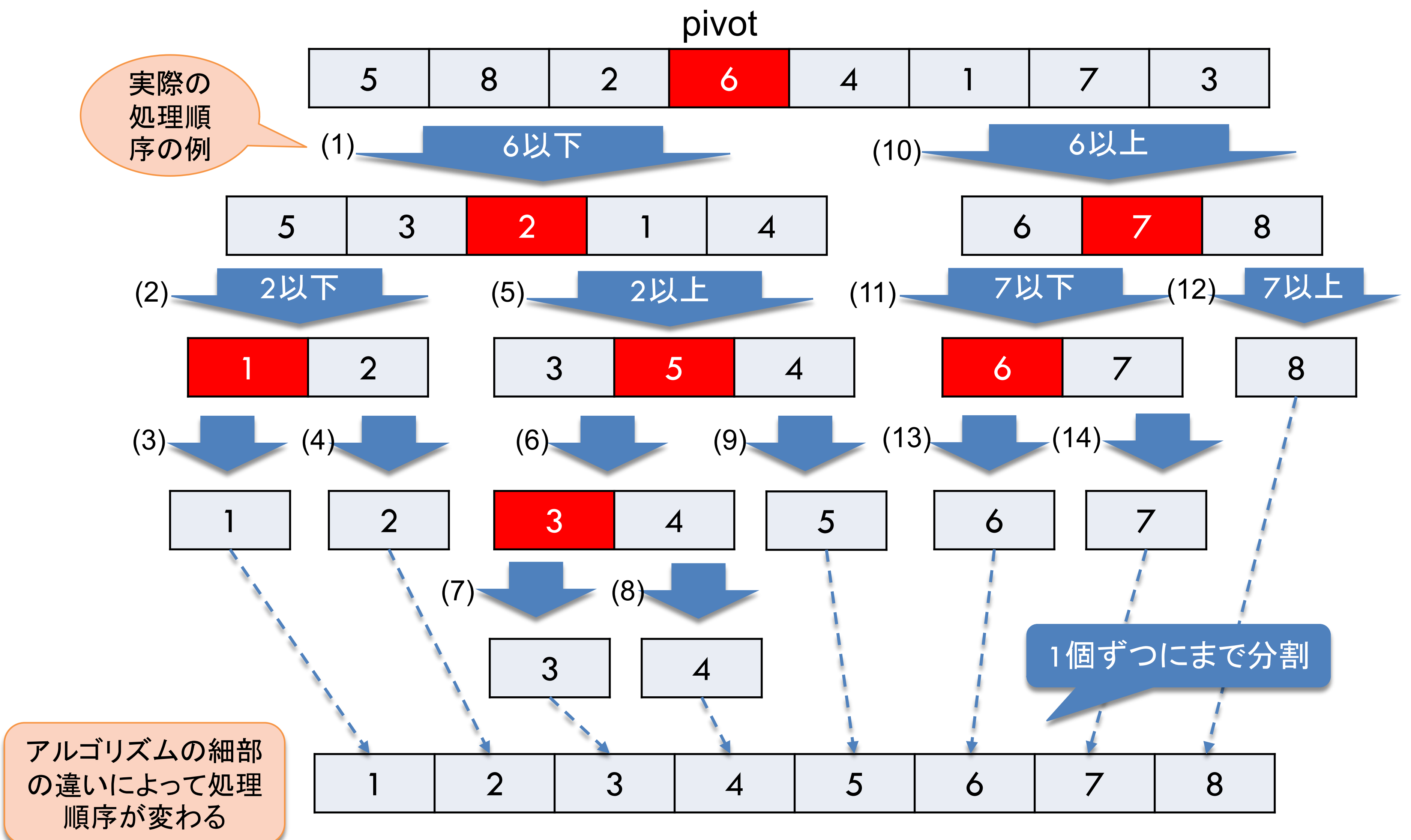
クイックソート

- アルゴリズム
  - まず配列の中の適当な要素をpivot（軸）に選ぶ
  - 配列の中のpivot以下の要素を左側，pivot以上の要素を右側に寄せて，配列を左右2つの部分配列に分割する
  - さらにそれぞれの部分配列に対しても，再帰的に同様の処理を適用して分割していく
  - すべての部分配列の要素が1つになるまで分割を繰り返すと，配列全体のソートが完了する
- 代表的な分割方法

「以下」と「以上」なのは
オーバーランを防ぐため



クイックソートの例

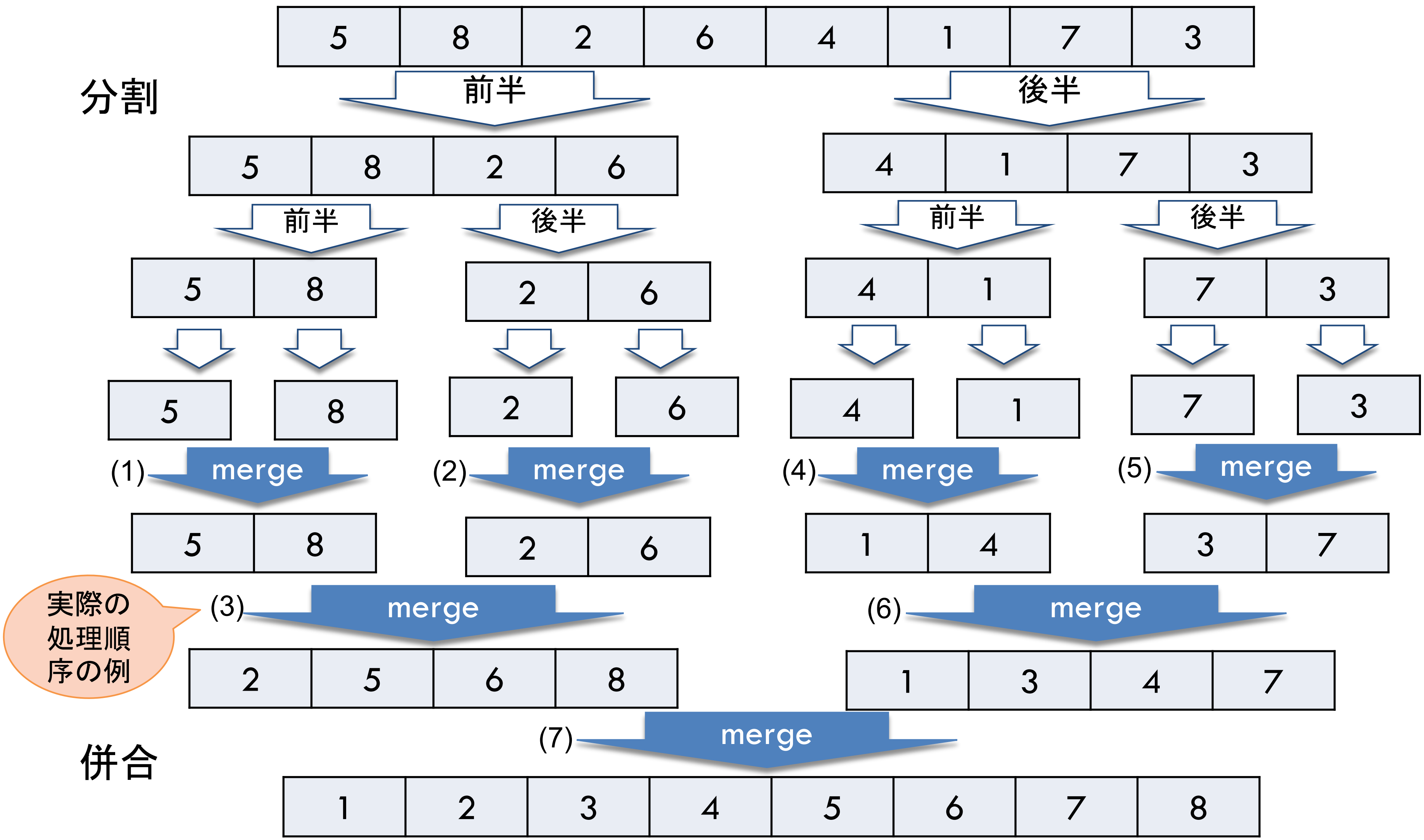

pivot
5 8 2 6 4 1 7 3
実際の処理順序の例
(1)
6以下
(10)
6以上
5 3 2 1 4
6 7 8
(2)
2以下
(5)
2以上
(11)
7以下
(12)
7以上
1 2
3 5 4
6 7
8
(3)
(4)
(6)
(9)
(13)
(14)
1
2
3 4
5
6
7
(7)
(8)
3
4
1個ずつにまで分割
アルゴリズムの細部の違いによって処理順序が変わる
1 2 3 4 5 6 7 8

マージソート

- アルゴリズム
  - 配列を要素数が半分ずつになるように，前半部分と後半部分に再帰的に分割していく
  - 要素が1つになるまで分割したものは，整列済みとみなせる
  - 分割とは逆に，整列済みの前半部分と後半部分を，再帰的にマージしていくと，配列全体のソートが完了する
  - ただし，前半部分と後半部分をマージして同じ領域に入れなおすには，前半部分を退避して空けておくことが必要になる

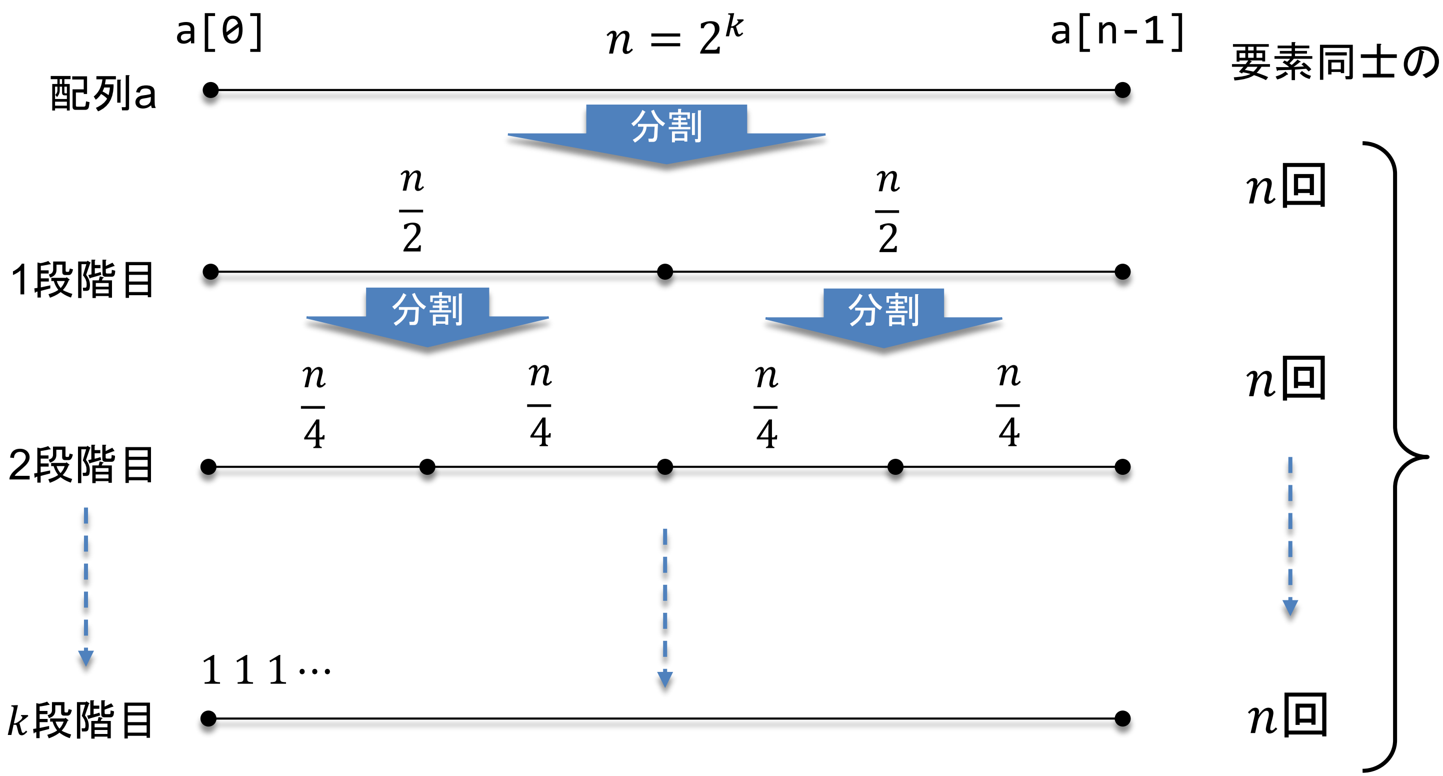
- 分割統治アルゴリズム
  - 再帰的に，対象をいくつかの部分に分割し，それらの部分にも同様な処理を適用し，結果を再結合していくアルゴリズム
  - クイックソート，マージソート，2分探索法など

マージソートの例


5 8 2 6 4 1 7 3
分割
前半
後半
5 8 2 6
4 1 7 3
前半
後半
前半
後半
5 8
2 6
4 1
7 3
5
8
2
6
4
1
7
3
(1) merge
(2) merge
(4) merge
(5) merge
5 8
2 6
1 4
3 7
実際の処理順序の例
(3) merge
(6) merge
2 5 6 8
1 3 4 7
(7) merge
併合
1 2 3 4 5 6 7 8

クイックソート/マージソートの計算量

理想的なクイックソートまたはマージソート



k段階目に各1個になるから $\dfrac{n}{2^k} = 1$

nに比例する回数の処理をk段階

計算量はnkに比例

nの式で表すと？

ソートアルゴリズムの特徴

- ソートの安定性
  - 同じ優先度の(キーが同じ)データの順序が維持されること
  - マージソートや前回の単純なソートは(順序が)安定である
  - クイックソートやシェルソートは高速だが(順序が)安定でない

- 内部ソート vs 外部ソート
  - 対象の配列の中で処理が完結するものを「内部ソート」という
  - マージソートは追加の記憶領域が必要な「外部ソート」である

- 比較ソート vs その他のソート
  - 通常のソートは要素同士をお互いに比較して交換していく
  - 要素の値(範囲)が少ない場合は，分類によってソートできる

補足：クイックソートの平均計算量

要素数がn個のときの平均比較回数をA_nと表すことにすると，以下のように考察される。

$$A_1 = 0$$

$$A_2 = 2 + A_1 + A_1 = 2 + 2A_1$$

$$A_3 = 3 + \frac{(A_2+A_1)+(A_1+A_2)}{2} = 3 + \frac{2}{2}(A_1 + A_2)$$

$$A_4 = 4 + \frac{(A_3+A_1)+(A_2+A_2)+(A_1+A_2)}{3} = 4 + \frac{2}{3}(A_1 + A_2 + A_3)$$

よって，A_nはまず以下のように表せる。

$$A_n = n + \frac{2}{n-1}\sum_{k=1}^{n-1} A_k$$

さらに，ここで nA_{n+1}と $(n-1)A_n$の式を縦に並べて引き算する。

$$nA_{n+1} - (n-1)A_n = (n+1)n - n(n-1) + 2A_n = 2n + 2A_n$$

この式を整理すると，漸化式を作れる。

$$nA_{n+1} = (n+1)A_n + 2n$$

$$\frac{A_{n+1}}{n+1} = \frac{A_n}{n} + \frac{2}{n+1}$$

A_1も考慮してA_n/nの一般項を求める。

$$\frac{A_n}{n} = 2\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right) - 2$$

ここで，nが十分に大きければ，

$$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \log n + \gamma$$

となることが知られているので（$\gamma \approx 0.577$），平均計算量は以下のように表される。

$$O(A_n) \approx O(n \log n)$$