アルゴリズム論
（第２回）

岩手県立大学
Iwate Prefectural University

佐々木研（情報システム構築学講座）
講師　山田敬三
k-yamada@iwate-pu.ac.jp

ソート

- レコード群について，あるキーの昇順や降順に並べ替えること
  - 非常に多くの場面で利用される.
- 内部ソート
  - メモリに記憶された配列をソート
- 外部ソート
  - 外部媒体へのアクセスを前提

ソーティングの計算量

- 計算量：アルゴリズムの
  効率性を評価する尺度
  - どれくらい手間がかかるかを評価
  - キーの比較回数Cとデータの置換回数Mが使用される.

- キーの比較回数が少ないものが高速
  - 置換回数は比較回数で抑えられる（C ≧ M）

- 一般的にはO（ビッグオー）記法が使用される.

単純選択法

岩手県立大学
Iwate Prefectural University

単純選択法

基本的な考え方

1. 全体で最小要素を選出して，
   それを左端に移動.

2. 次にそれを除いた残りの中から
   最小要素を選出. . .

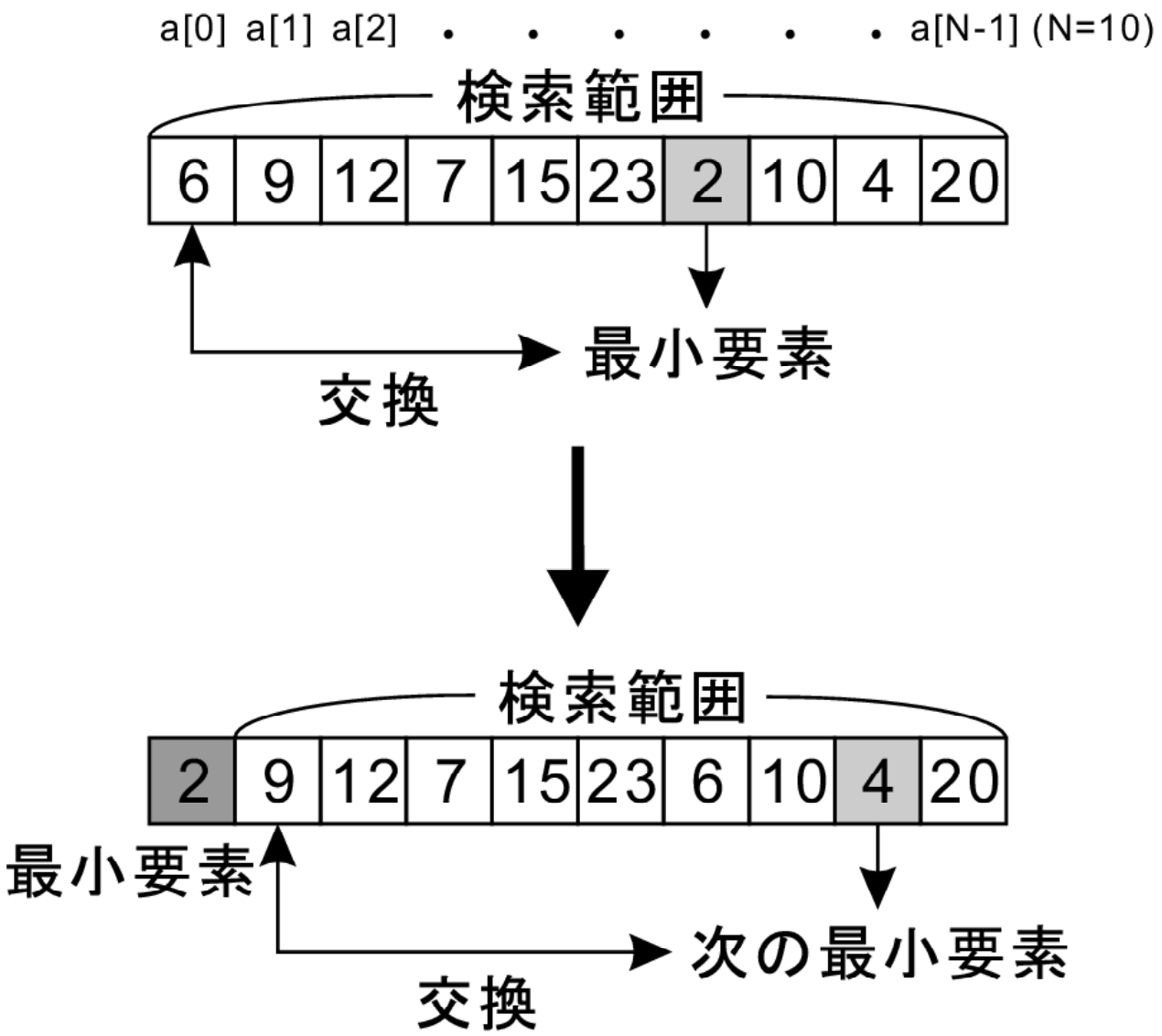
単純選択法

具体的には

- 配列番号をiとする.
  （検索後の最小要素を入れる.）

  以下を**i=0～N-2**について繰り返し
- **a[i]～a[N-1]**の中から最小要素**a[k]**を検索し，**a[i]**と交換

単純選択法


a[0] a[1] a[2] a[N-1] (N=10)
検索範囲
6 9 12 7 15 23 2 10 4 20
最小要素
交換
検索範囲
2 9 12 7 15 23 6 10 4 20
最小要素
次の最小要素
交換

実行例

i=0	6	9	12	7	15	23	2	10	4	20
i=1	2	9	12	7	15	23	6	10	4	20
i=2	2	4	12	7	15	23	6	10	9	20
i=3	2	4	6	7	15	23	12	10	9	20
i=4	2	4	6	7	15	23	12	10	9	20
i=5	2	4	6	7	9	23	12	10	15	20
i=6	2	4	6	7	9	10	12	23	15	20
i=7	2	4	6	7	9	10	12	23	15	20
i=8	2	4	6	7	9	10	12	15	23	20
	2	4	6	7	9	10	12	15	20	23

単純選択法の計算量

- キー比較回数は初期状態に影響されない

$$C = (N-1) + (N-2) + \ldots + 1 = \frac{1}{2}(N^2 - N)$$

- データ置換回数

$$M = N - 1$$

バブルソート法

岩手県立大学
Iwate Prefectural University

バブルソート法

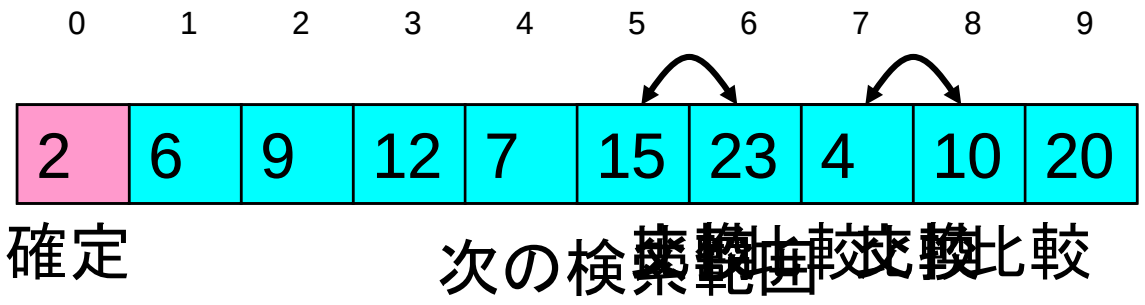
基本的な考え方

1. 隣の要素と比較する.

2. 順序が入れ替わっていれば，順序を正す.

バブルソート法

1. i=0～N-2について以下を繰り返す
2. jをN-1とする
3. a[j]とa[j-1]を比較し、a[j]が小さければ交換
4. jを1減らし、j>iのとき3.にもどる



バブルソート法の計算量

比較回数

- 配列状態に関わらず同じ

$$C = \frac{1}{2}(N^2 - N)$$

バブルソート法の計算量

置換回数

- 正順に並んでいる場合が最小

$$M_{\min} = 0$$

- 逆順に並んでいる場合が最大

$$M_{\max} = \frac{1}{2}(N^2 - N)$$

- 平均は

$$M_{\text{ave}} = \frac{1}{4}(N^2 - N)$$

クイックソート法

岩手県立大学
Iwate Prefectural University

クイックソート法

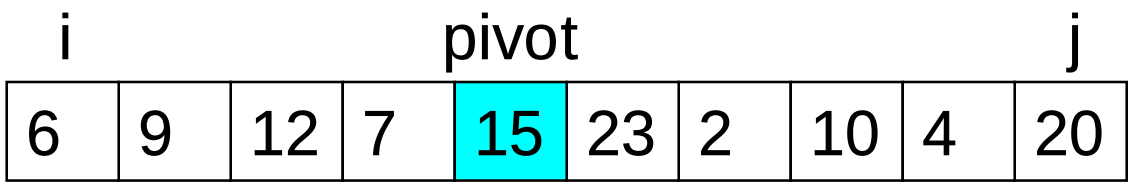
考え方

- 与えられた問題を，繰り返しより小さな問題に分割して，その小問題を解く（分割統治法）

- 例）解答用紙を整列する：まず十の位でソートして、できた各束を一の位でソートする.

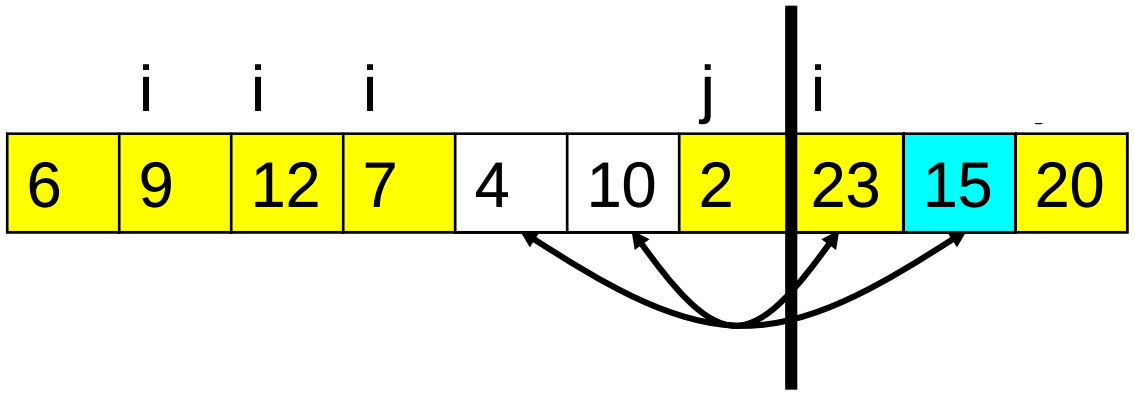
クイックソート法

1. iを先頭要素番号、jを末尾要素番号とする
2. pivotを配列の任意の要素から設定
3. i>jとなるまで4.～6.を繰り返す



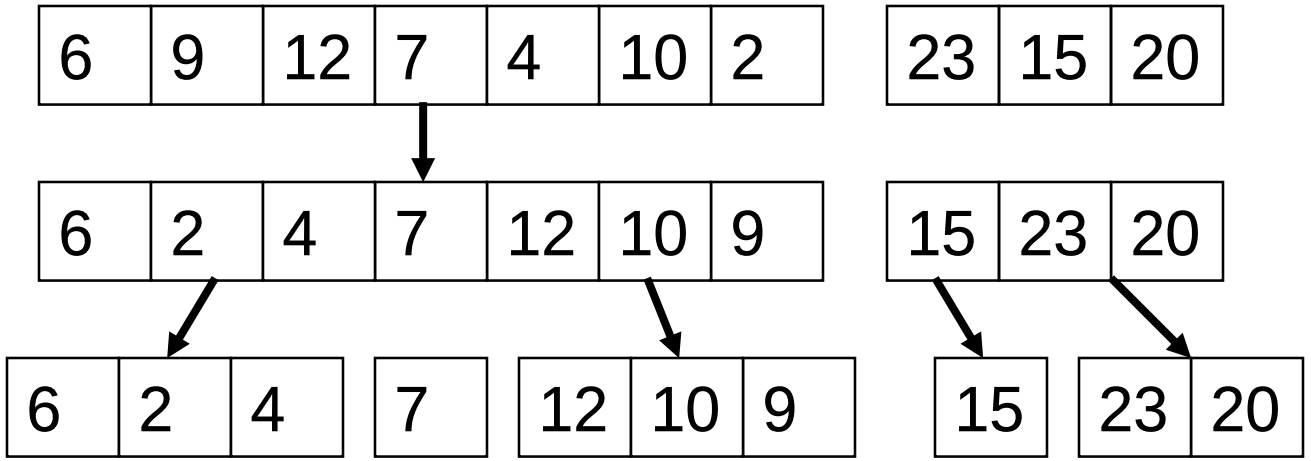
クイックソート法

4. iを増加させ、a[i]≧pivotが見つかるまで走査
5. jを減少させ、a[j]≦pivotが見つかるまで走査
6. i≦jならばa[i]とa[j]を交換し、iを1増加、jを1減少させて3.へ



クイックソート法

- 分割された各部分に対して再び**1.～6.**の手順を繰り返す
- 分割した結果、対象となる要素数が1となったら終了



クイックソート法の計算量

- 平均 $O(N \log N)$
- 最大 $O(N^2)$

A= 2

N	1	logN	N^A	A^N
2	1	1	4	4
4	1	2	16	16
8	1	3	64	256
16	1	4	256	65,536
100	1	7	10,000	1.27E+30
1,000	1	10	1,000,000	1.07E+301