第4回のキーワード

アルゴリズム関係

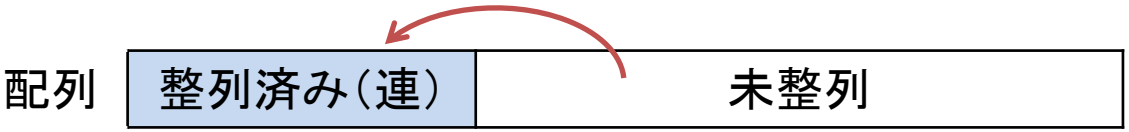
- ソート，並び替え，整列
- バブルソート/単純交換法（bubble sort）
- 選択ソート/単純選択法（selection sort）
- 挿入ソート/単純挿入法（insertion soft）
- 連（run）
- $O(n^2)$
- マージ（merge）

Java関係

- 配列の要素を交換する
- 配列から最小値を選ぶ
- 配列の要素をずらす
- a[i++]

単純なソートの基本戦略

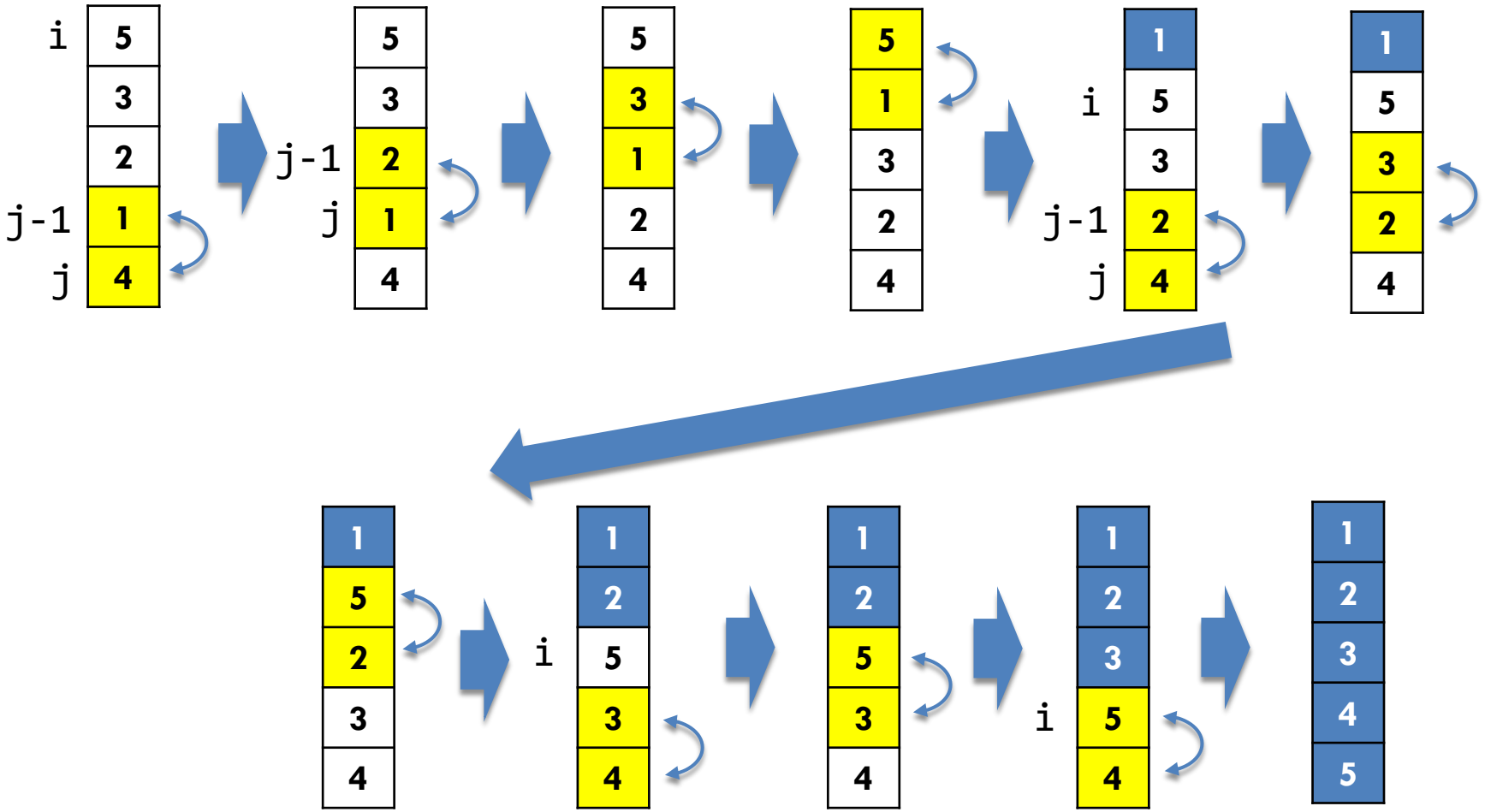
- 「連」を成長させる
  - 配列の前半に整列済みの列(「連」という)を作り，残りの未整列の部分から1つずつ要素を移して，連を伸ばしていく
  - そのために使う基本アルゴリズム：配列の要素を交換する / 配列から最小値を選ぶ / 配列の要素をずらす



- バブルソート
  - 未整列の範囲の末尾から先頭に向かって，順にとなり合う要素を比較し，小さい順でなければ要素を交換していく
  - すると，最小値が先頭に来るので，整列済みの範囲を1要素分広げ，残りの未整列の範囲に対して同様の処理を行う
  - 以上の手順を繰り返すと，最終的に全範囲が整列済みになる

バブルソートの例

- となり同士を比較して交換していく



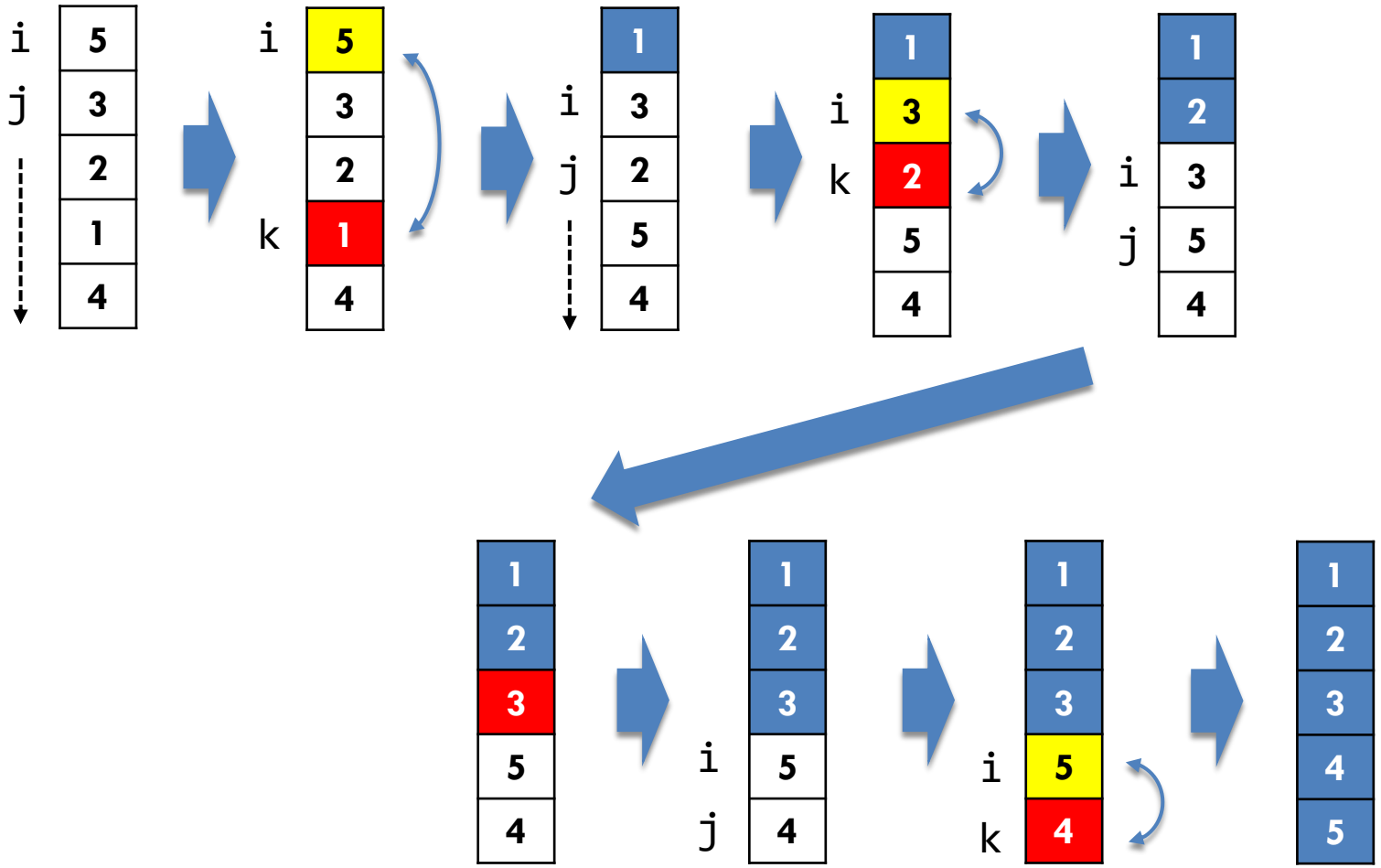
少しましなソート

- 選択ソート
  - 未整列の範囲の要素を順に調べて最小の要素を選択し，未整列の範囲の先頭の要素と交換する
  - すると，整列済みの範囲が1要素分広がるので，残りの未整列の範囲に同様の処理を行う
  - 以上の手順を繰り返すと，最終的に全範囲が整列済みになる

- 挿入ソート
  - まず，配列の先頭の要素は整列済みとする
  - 未整列の範囲の先頭の要素を取り出すとその位置は空く
  - 空きを利用して整列済みの要素を末尾から1つずつ後ろにずらすことで，取り出した要素を大小関係が適切な位置に挿入する
  - 以上の手順を繰り返すと，最終的に全範囲が整列済みになる

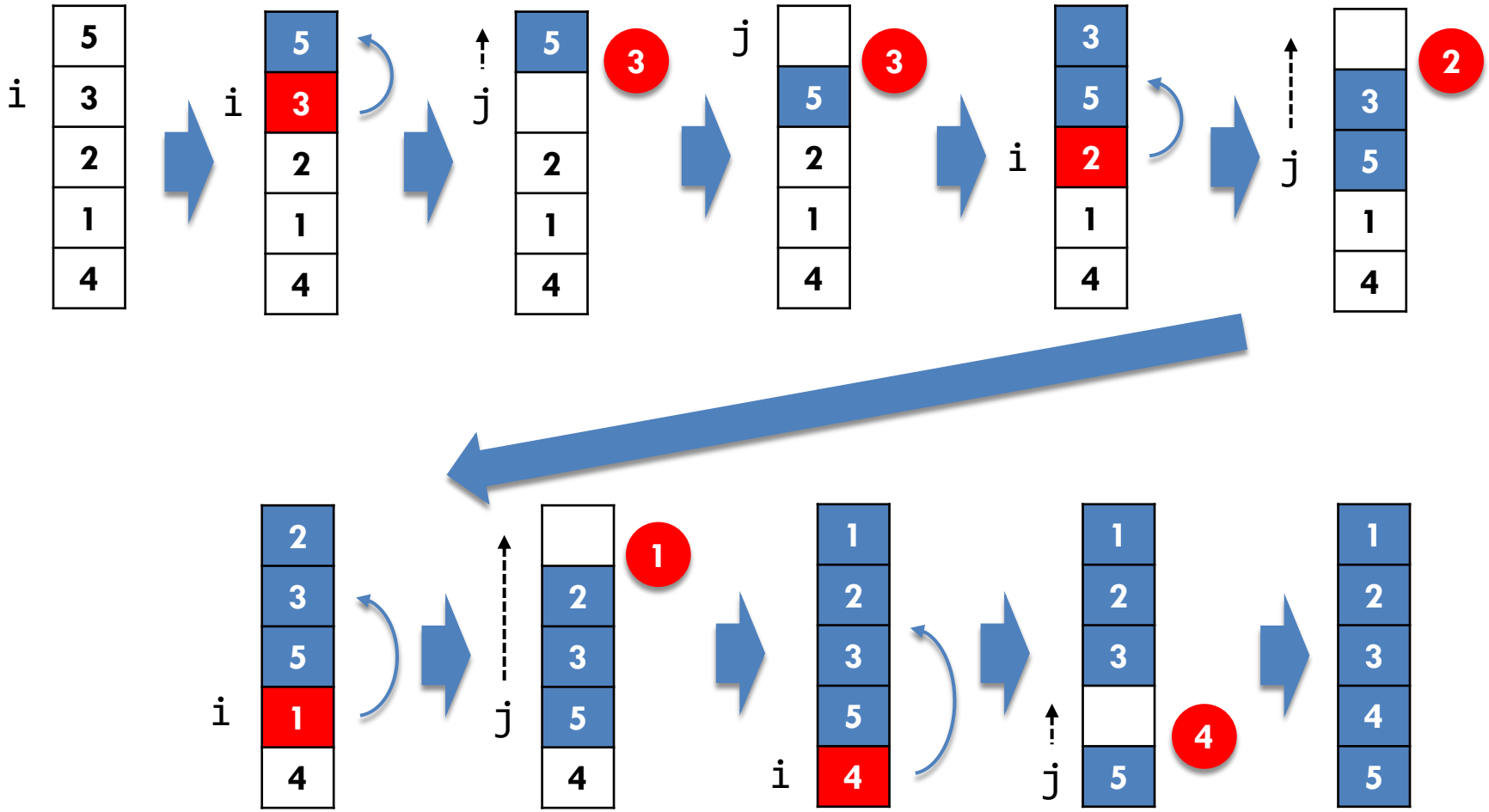
選択ソートの例

- 未整列範囲から最小値を選択して並べていく



挿入ソートの例

- 未整列の値を整列済み範囲に挿入していく



単純なソートの計算量

- 比較と交換
  - ソートでは，要素同士の比較回数 > 要素同士の交換回数
  - 計算オーダーについては，比較回数だけを考慮すればよい
  - ただし，交換の方が時間がかかるので実用上は注意が必要
- バブルソートと選択ソート
  - ともに2重ループを全部回るので，比較回数は $n(n-1)/2$
  - その上で，交換回数が劇的に少ない選択ソートの方が高速
- 挿入ソート
  - 最悪の場合は，ループを全部回り，比較回数は $n(n-1)/2$
  - 最善の場合は，n 回の比較だけで全く要素を交換せずに完了

配列のマージ

- 整列済みの配列を併合する
  - 先頭同士を比較して，先に来るものを取り出して並べる
  - ただし，配列の中に残りがない場合に注意する