

# ソートアルゴリズム

d-hacks B3 mioto

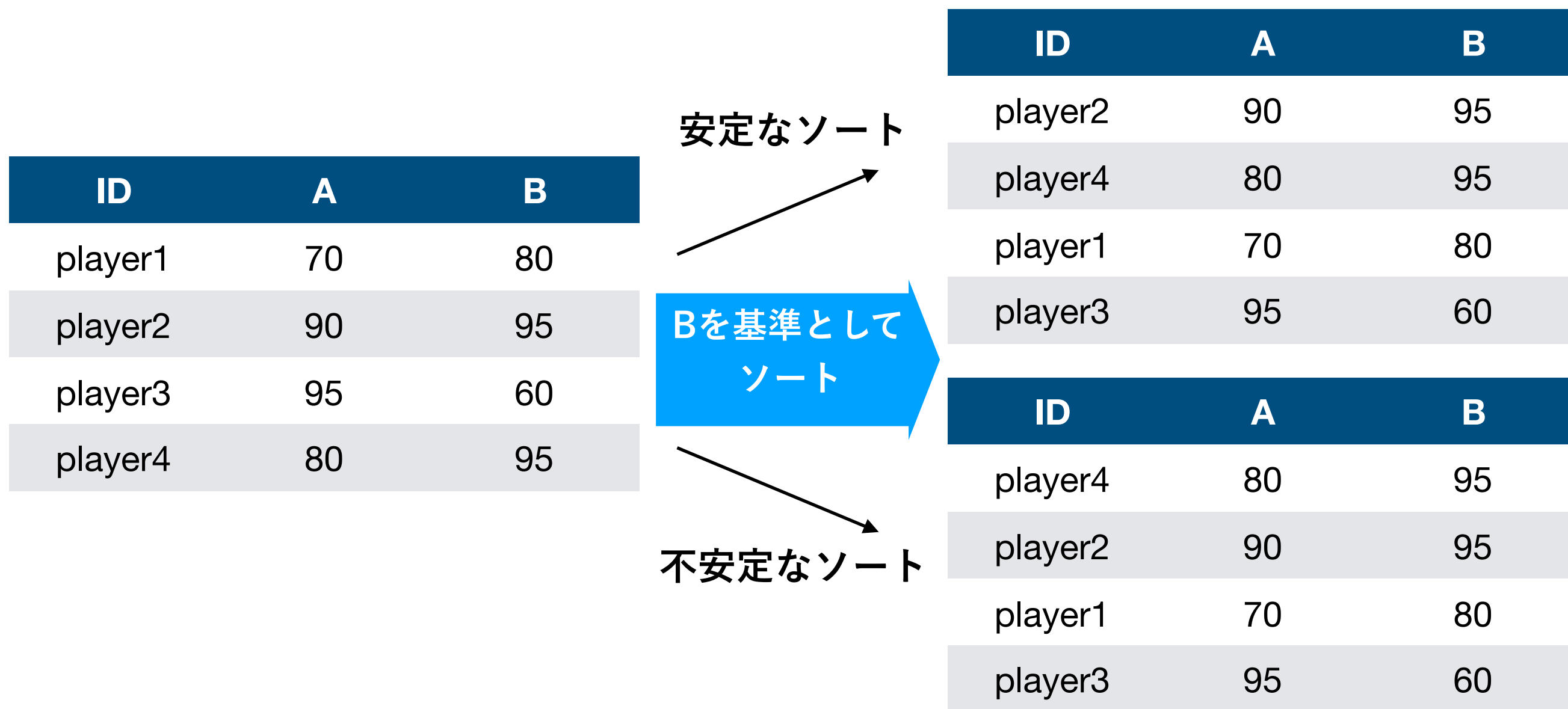
# データの属性とソート

- データ = 複数の属性を持つ表
- キー：属性（ID, A, B）
- ソート（整列）
  - あるキーを基準にデータを大きい・小さい順に並び替える。

ID	A	B
player1	70	80
player2	90	95
player3	95	60
player4	80	95

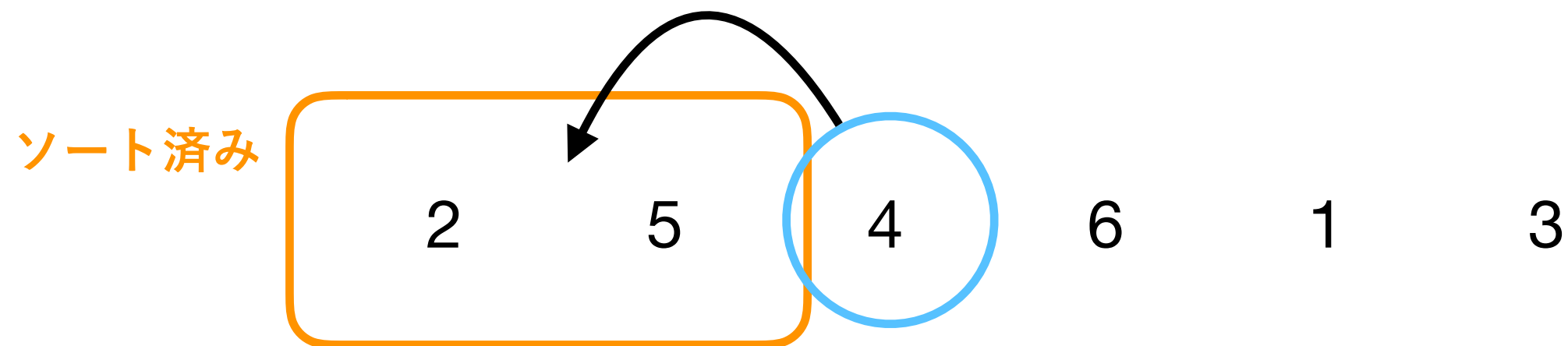
# ソートの安定性

- 安定なソート
  - キーの値が同じ要素を複数含む場合ソートの前後で順序が変化しない



# 挿入ソート

- 端から1枚ずつカードを取り出す
- その時点で既にソート済みの並びの適当な位置に挿入
- 安定なソートアルゴリズム
- $O(N^2)$

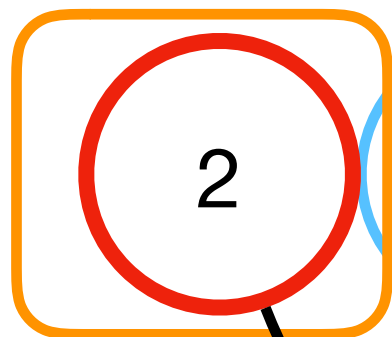


# 挿入ソート

ソート済み

$v=2$

$j$



5

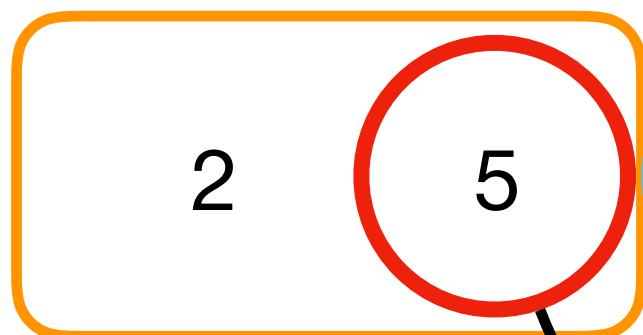
4

6

1

3

$i$



2

5

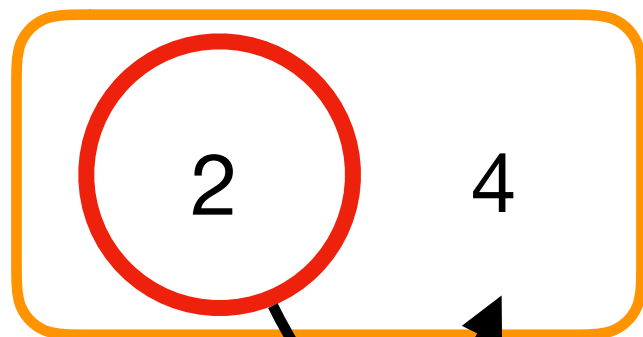
4

6

1

3

$v=4$



2

4

5

6

1

3

# 挿入ソート

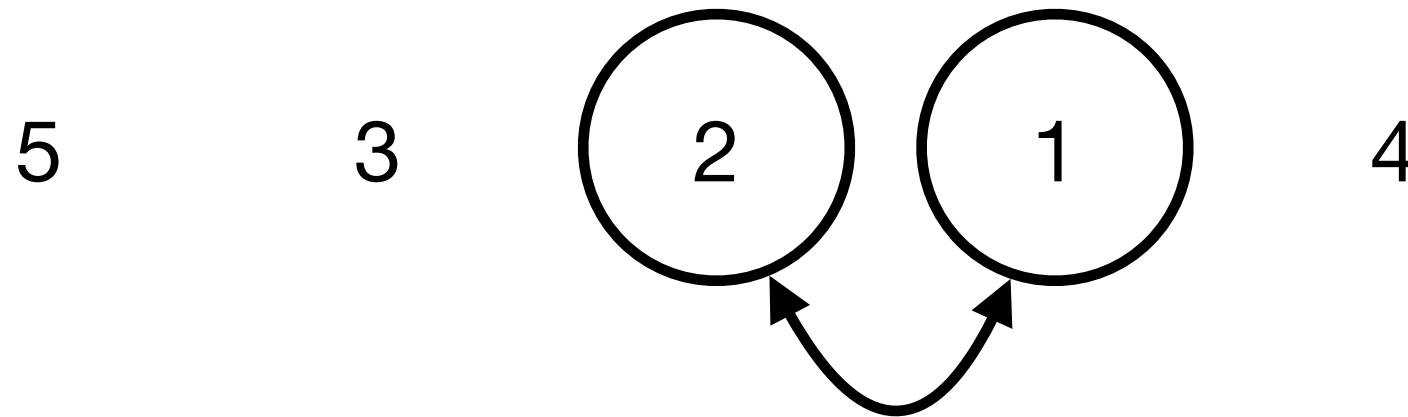
2      4      5      6      1      3

1      2      4      5      6      3

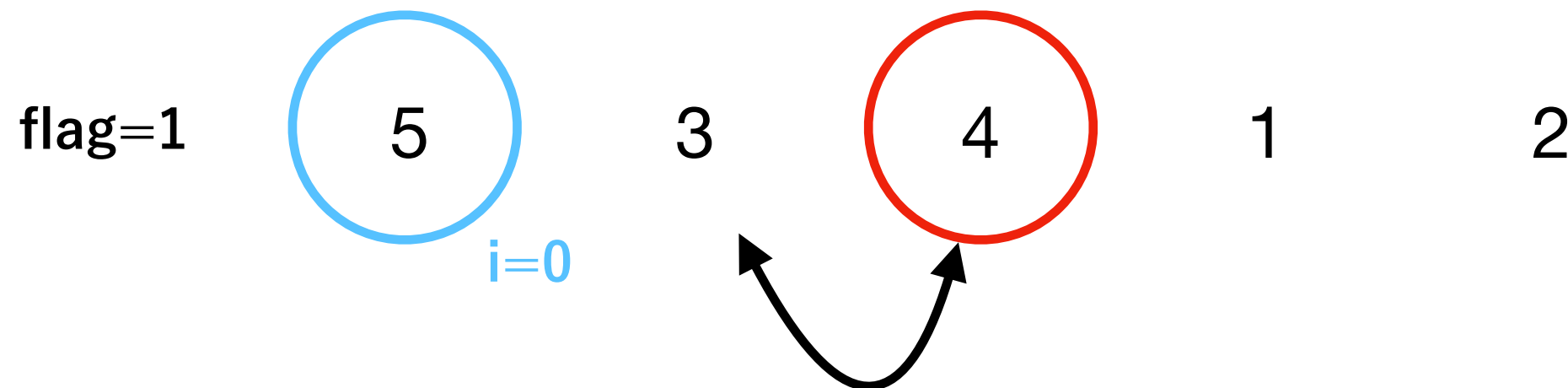
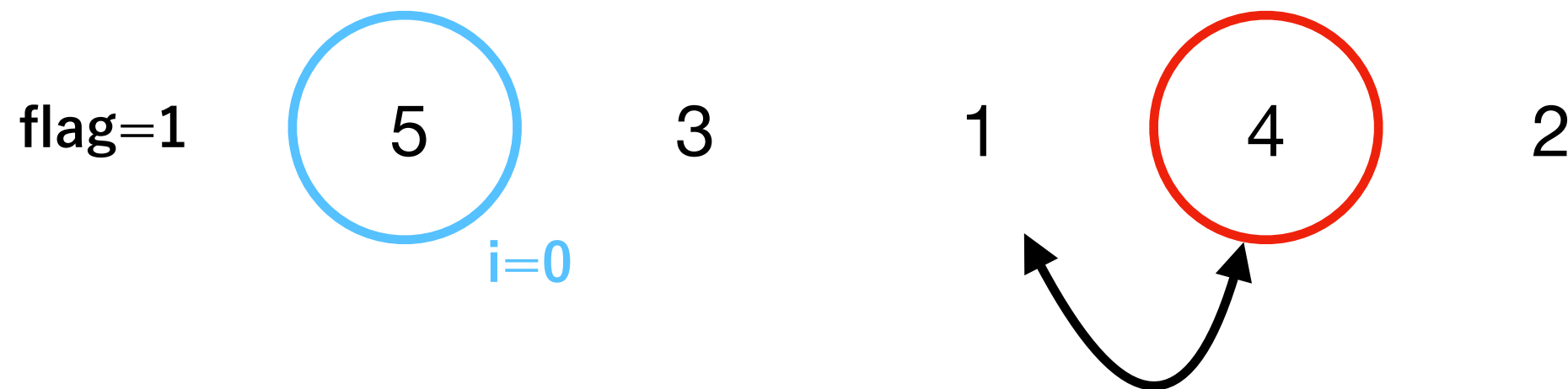
1      2      3      4      5      6

# バブルソート

- 隣り合った数を順序に従うように交換する
- 安定なソートアルゴリズム
- $O(N^2)$

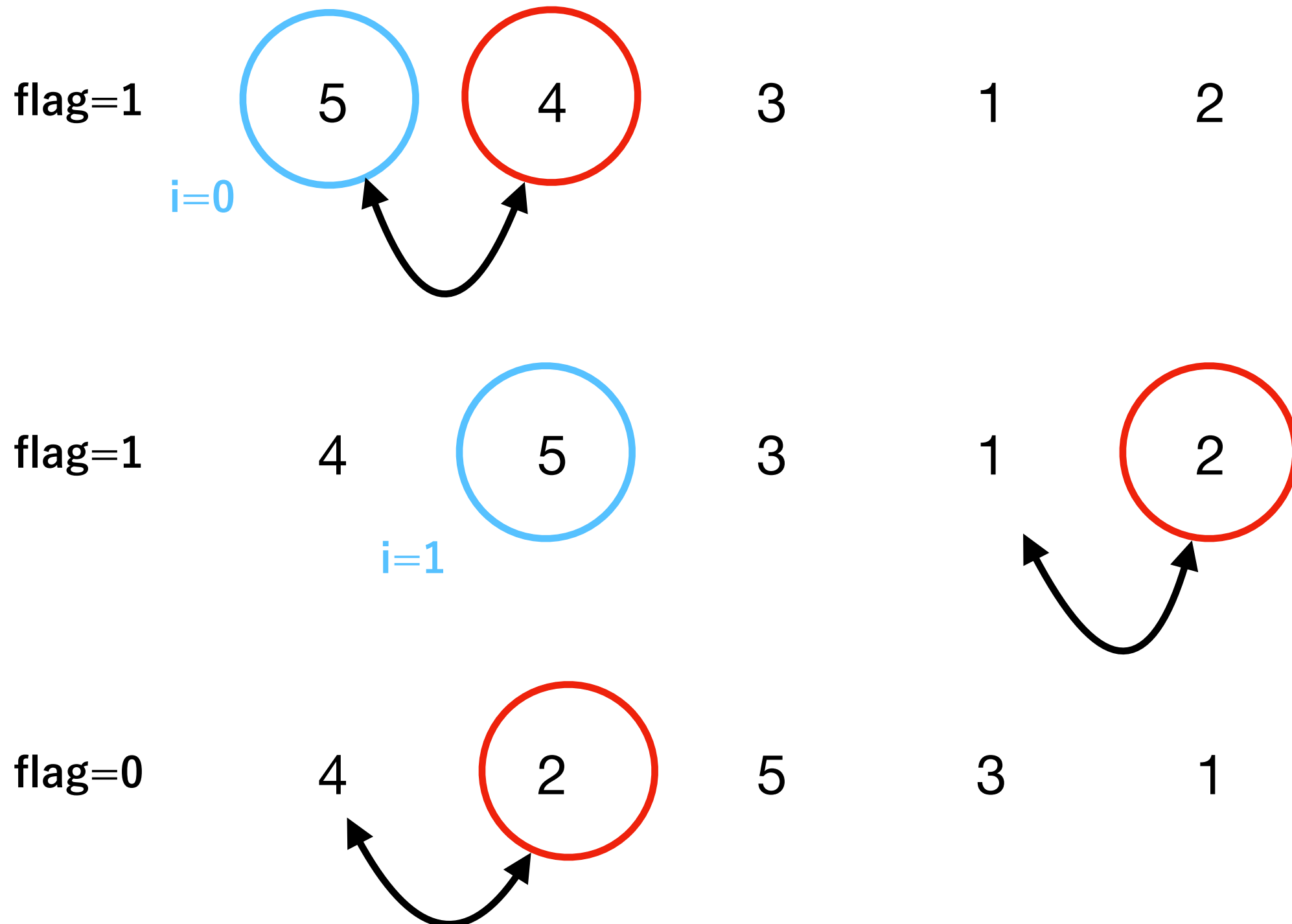


# バブルソート



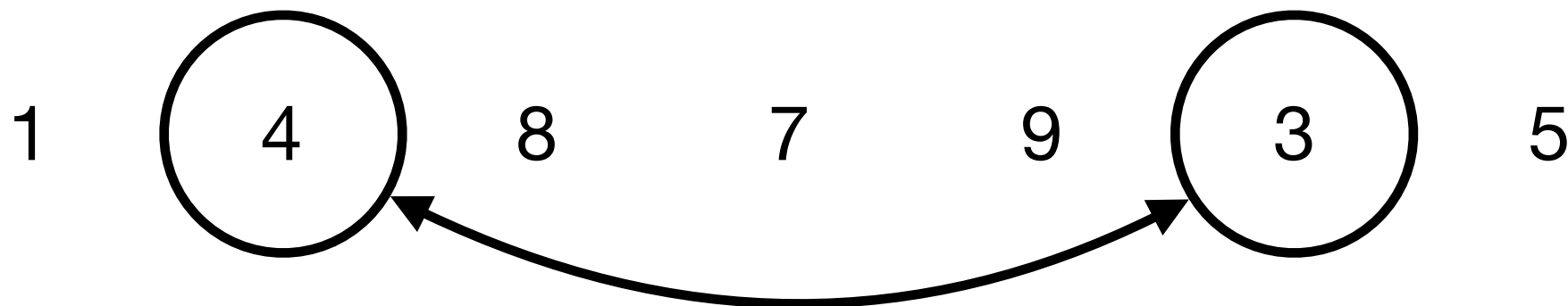
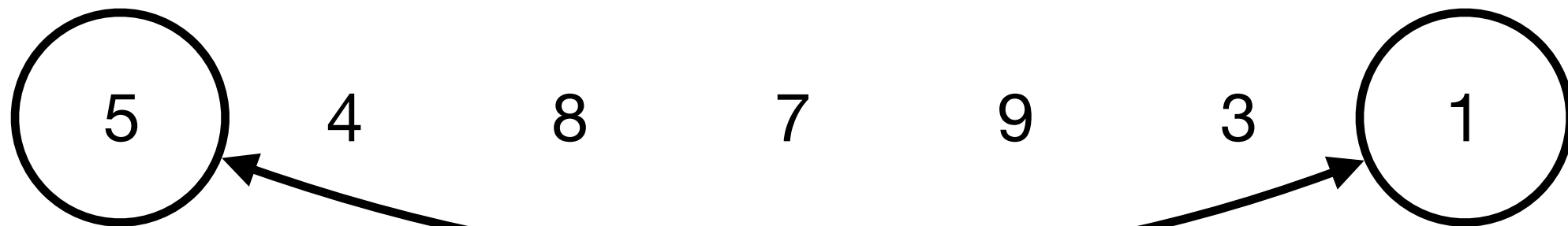


# バブルソート



# 選択ソート

- 未ソートのベクトルの内最小の値を選択
- 不安定なソートアルゴリズム
- $O(N^2)$



# 安定なソート

- バブルソート
- 選択ソート

# シェルソート

- 挿入ソート
- シェルソート