programmers

정렬

코딩테스트 광탈방지 A to Z : JavaScript - 이선협 @kciter

만약 구슬들을 크기 별로 나열해야 한다면?

정렬

요소들을 일정한 순서대로 열거하는 알고리즘

정렬의 특징

- 정렬 기준은 사용자가 정할 수 있다.
- 크게 비교식과 분산식 정렬로 나눌 수 있다.
- 대부분의 언어가 빌트인으로 제공해준다.
- 삽입, 선택, 버블, 머지, 힙, 퀵 정렬 등 다양한 정렬 방식이 존재한다.

어떤 정렬이 제일 빠를까?

Sorting Algorithms Animations

The following animations illustrate how effectively data sets from different starting points can be sorted using different algorithms.

https://www.toptal.com/developers/sorting-algorithms

비교식 정렬

버블 정렬

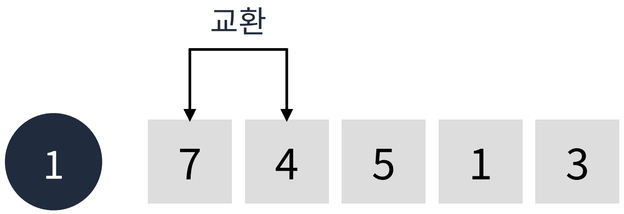
서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.

버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.

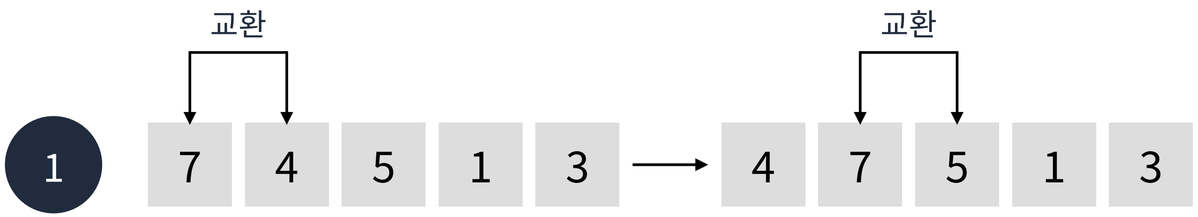
버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



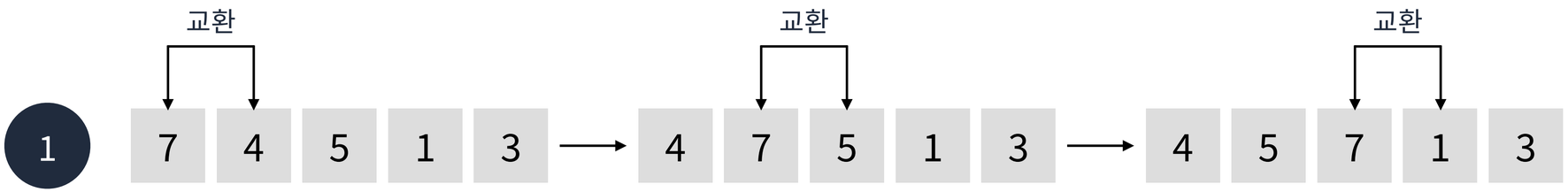
버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



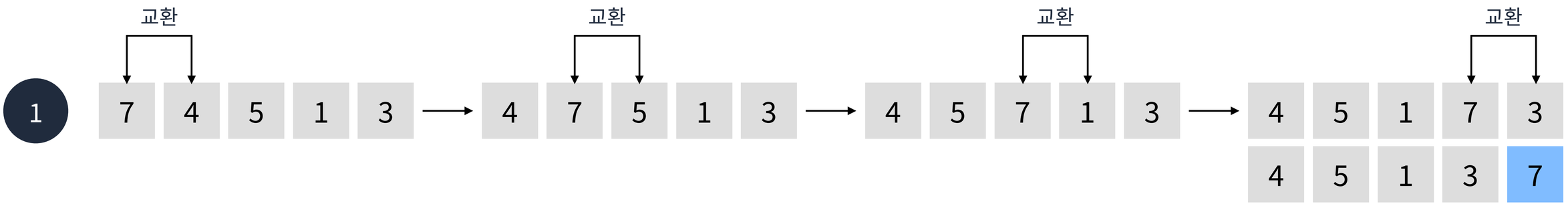
버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



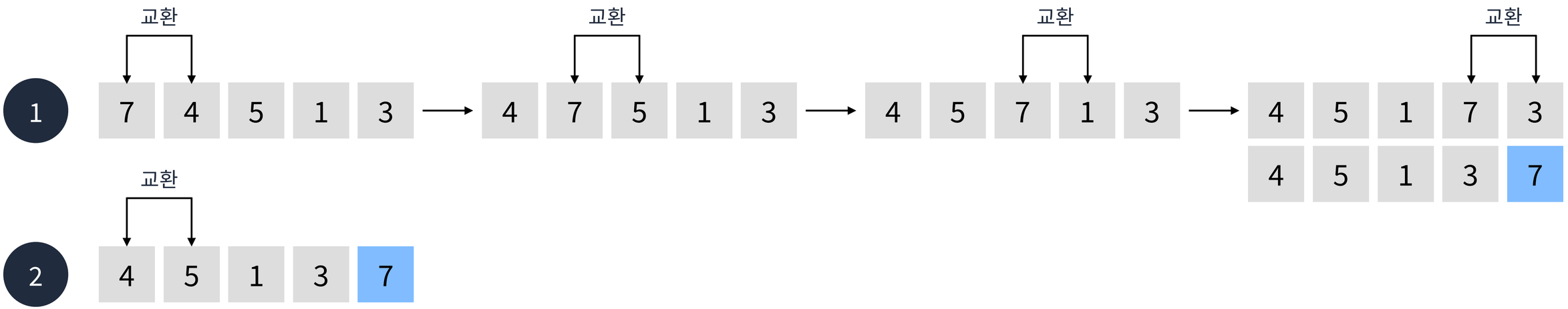
버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



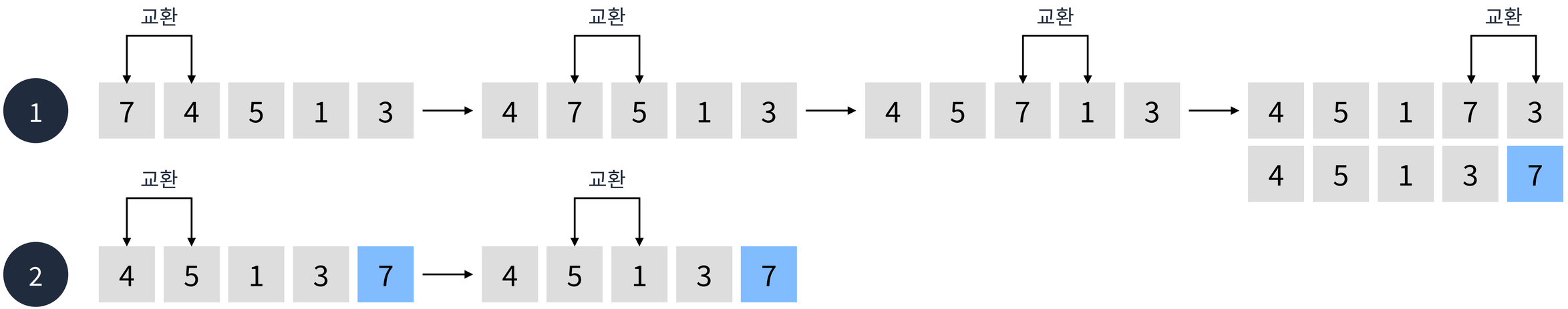
버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



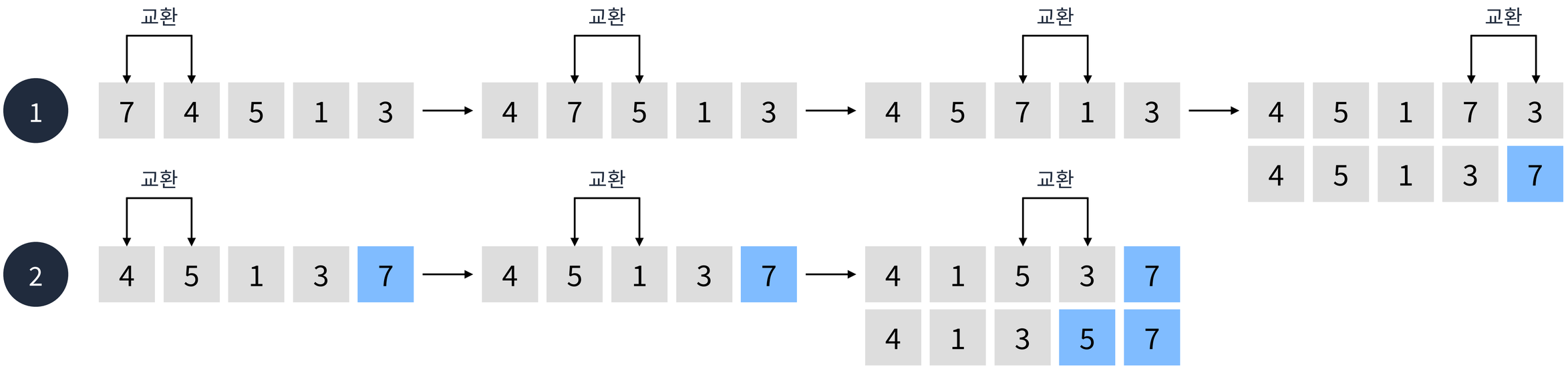
버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



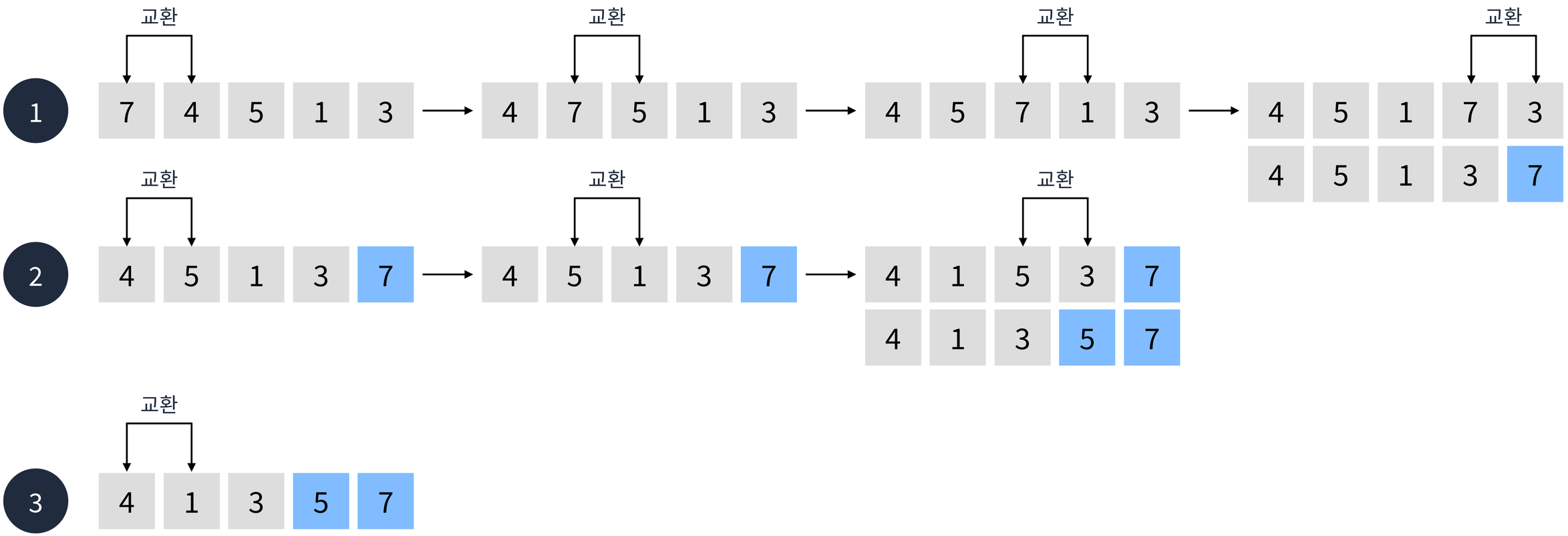
버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



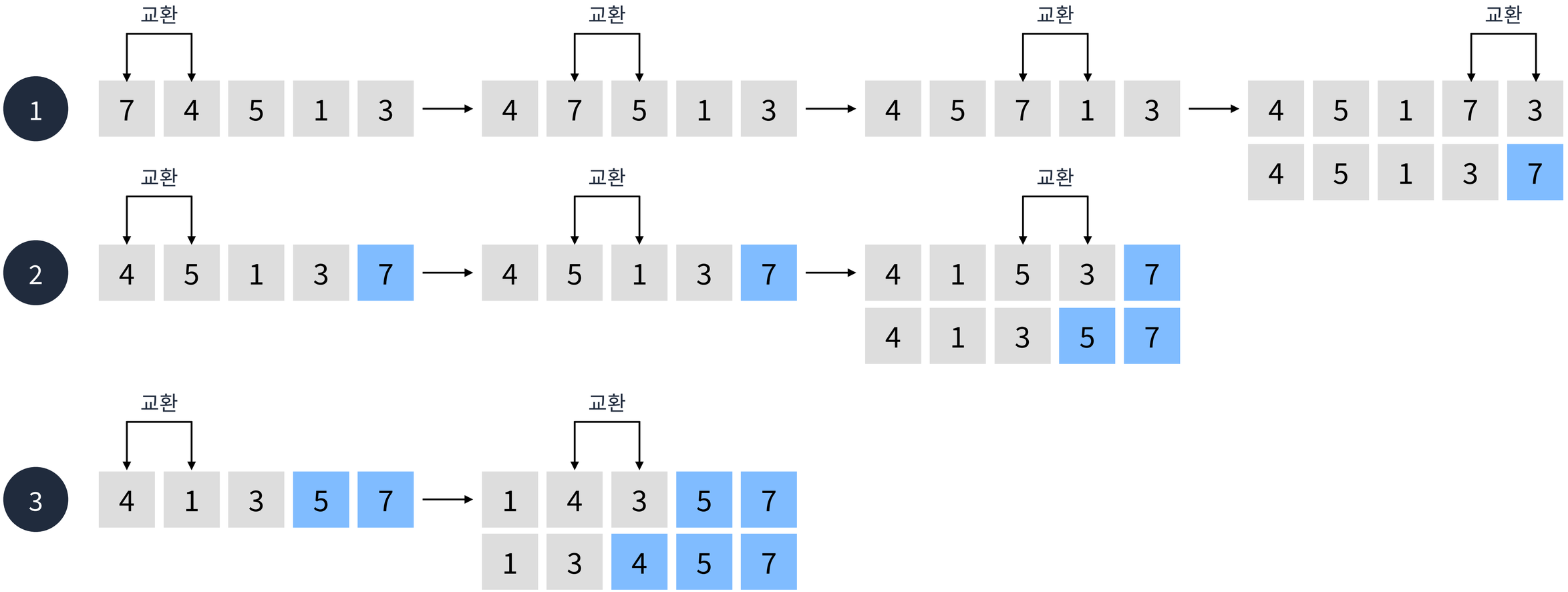
버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



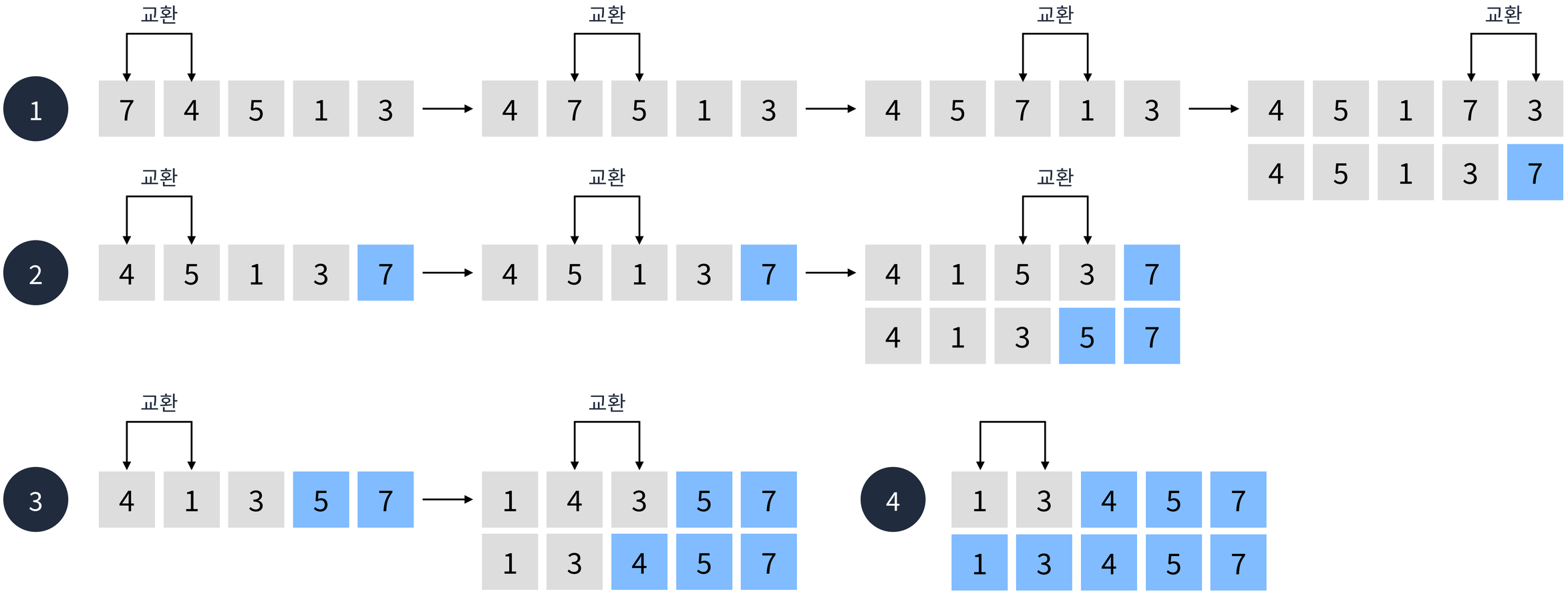
버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



버블 정렬

서로 인접한 두 요소를 검사하여 정렬하는 알고리즘
$O(n^2)$ 시간복잡도를 가진다.

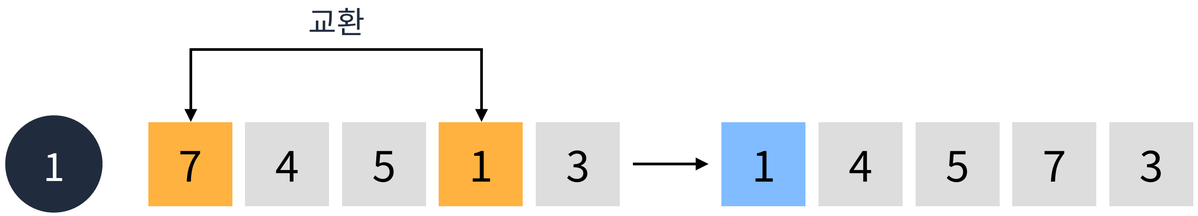


선택 정렬

선택한 요소와 가장 우선순위가 높은 요소를 교환하는 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.

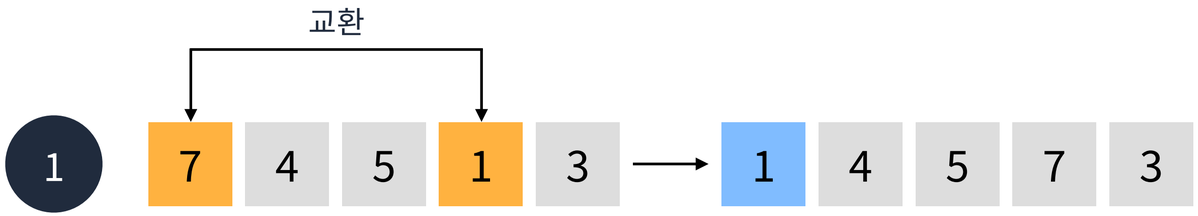
선택 정렬

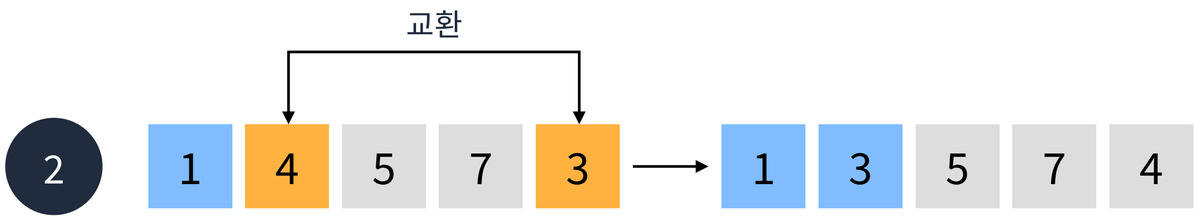
선택한 요소와 가장 우선순위가 높은 요소를 교환하는 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



선택 정렬

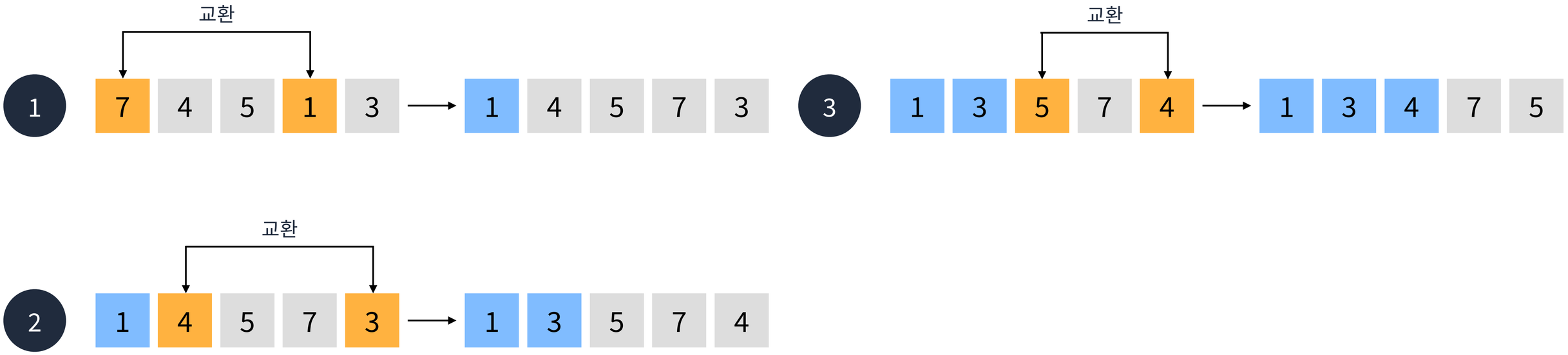
선택한 요소와 가장 우선순위가 높은 요소를 교환하는 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.





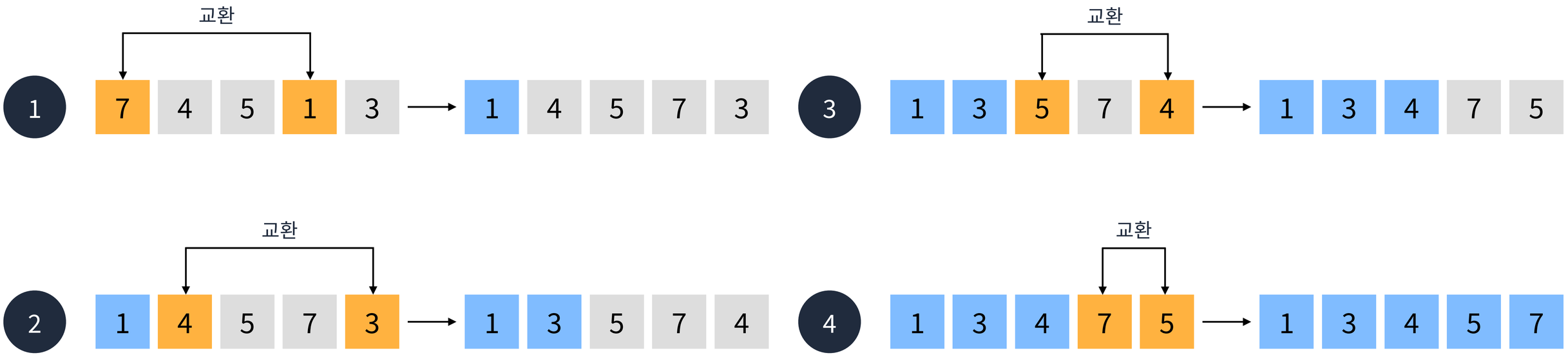
선택 정렬

선택한 요소와 가장 우선순위가 높은 요소를 교환하는 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



선택 정렬

선택한 요소와 가장 우선순위가 높은 요소를 교환하는 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.

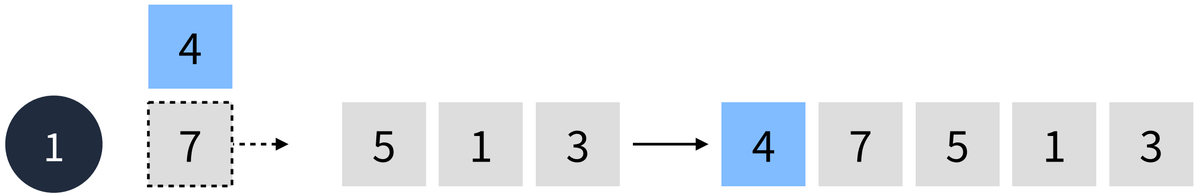


삽입 정렬

선택한 요소를 삽입 할 수 있는 위치를 찾아 삽입하는 방식의 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.

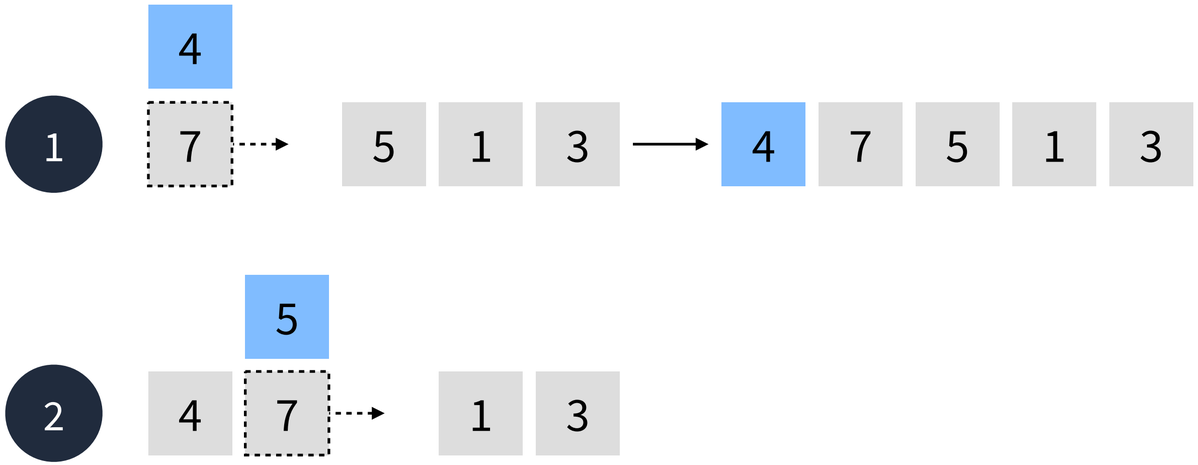
삽입 정렬

선택한 요소를 삽입 할 수 있는 위치를 찾아 삽입하는 방식의 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



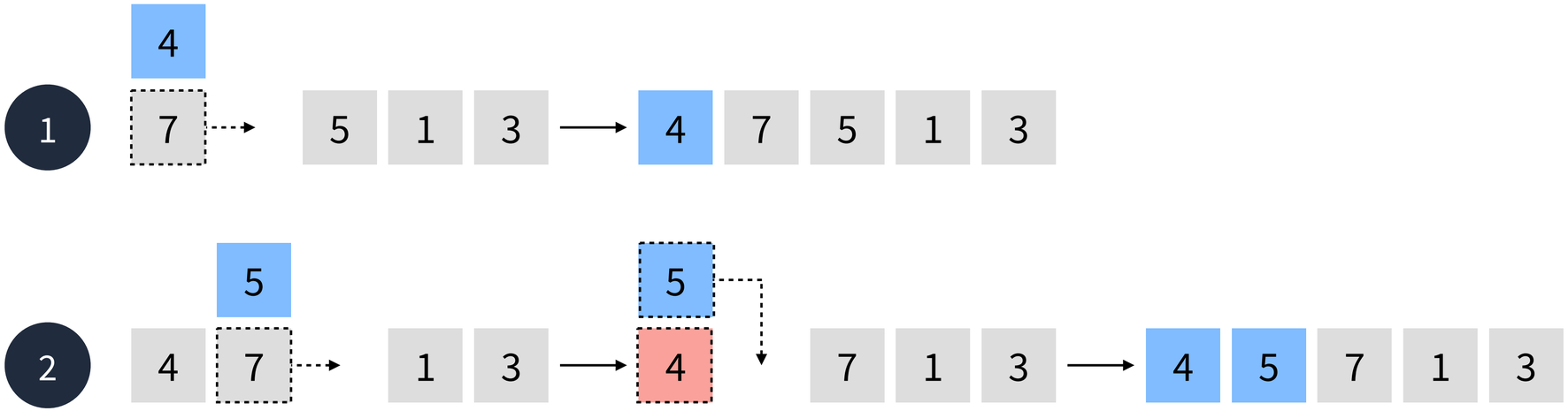
삽입 정렬

선택한 요소를 삽입 할 수 있는 위치를 찾아 삽입하는 방식의 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



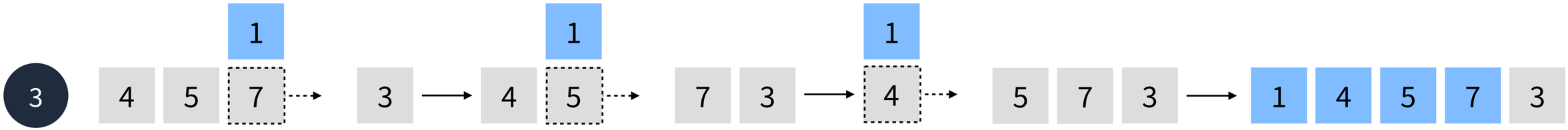
삽입 정렬

선택한 요소를 삽입 할 수 있는 위치를 찾아 삽입하는 방식의 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



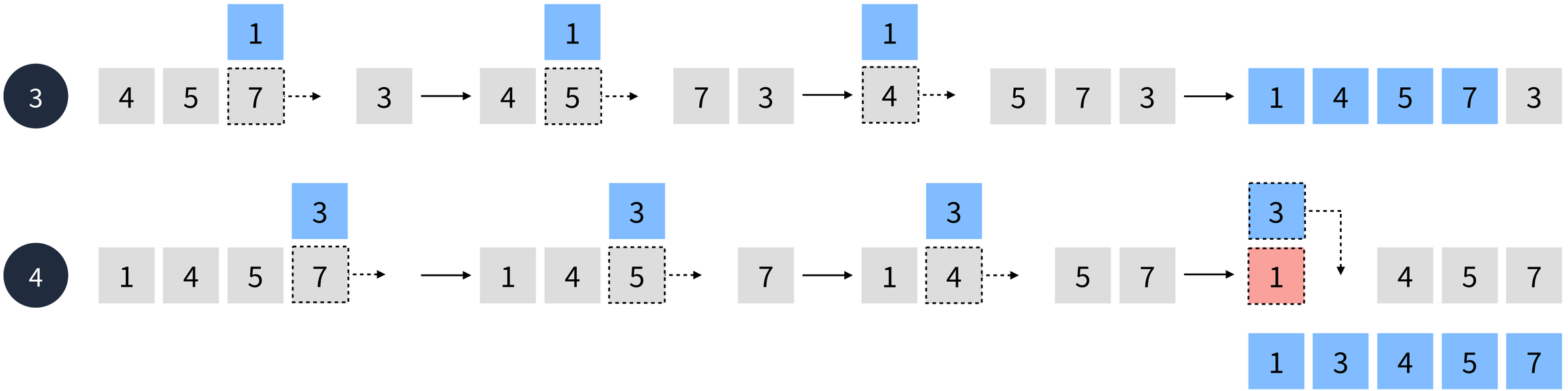
삽입 정렬

선택한 요소를 삽입 할 수 있는 위치를 찾아 삽입하는 방식의 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



삽입 정렬

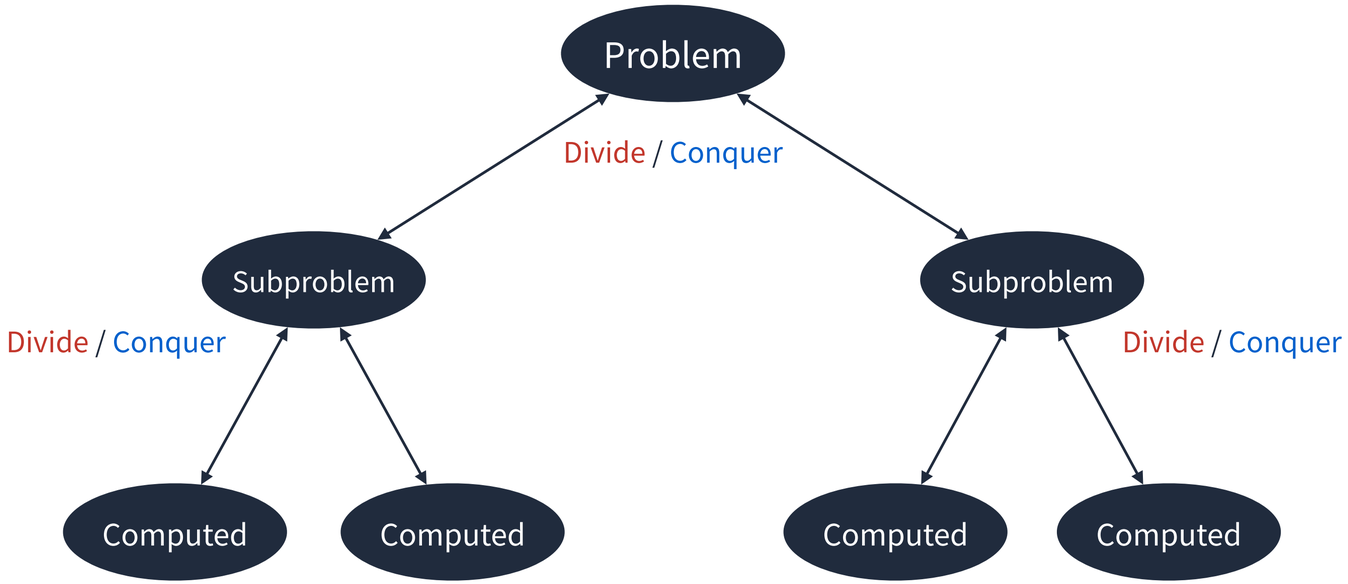
선택한 요소를 삽입 할 수 있는 위치를 찾아 삽입하는 방식의 정렬 알고리즘
$O(n^2)$ 시간복잡도를 가진다.



분산식 정렬

분할 정복

문제를 작은 2개의 문제로 분리하고 더 이상 분리가 불가능 할 때 처리한 후 합치는 전략
다양한 알고리즘에 응용된다.



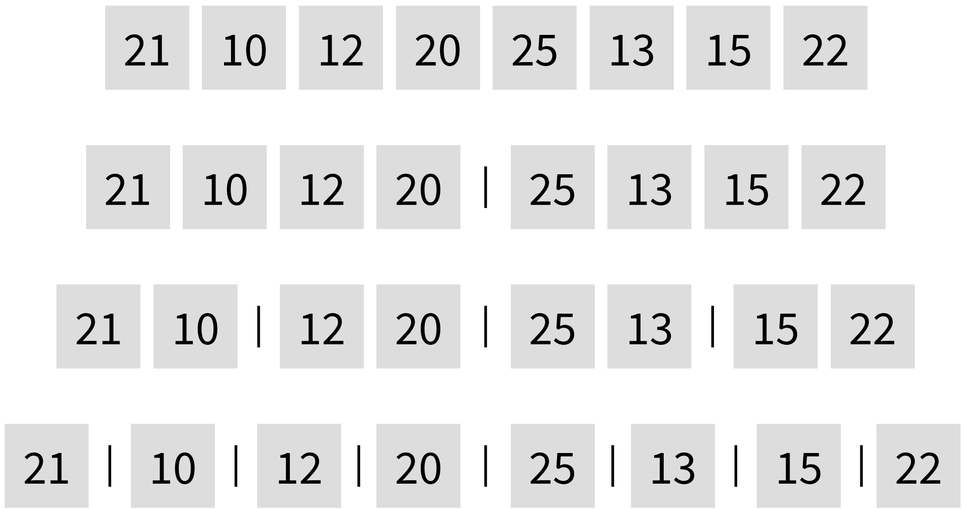
합병 정렬

분할 정복 알고리즘을 이용한 최선과 최악이 같은 안정적인 정렬 알고리즘
O(n log n) 시간복잡도를 가진다.

합병 정렬

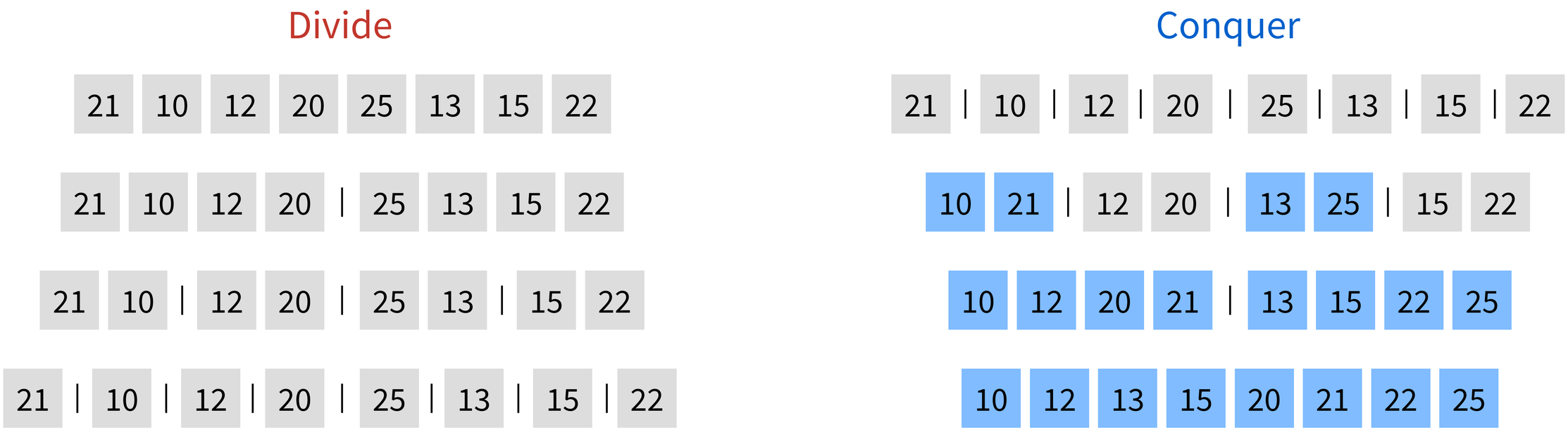
분할 정복 알고리즘을 이용한 최선과 최악이 같은 안정적인 정렬 알고리즘
O(n log n) 시간복잡도를 가진다.

Divide



합병 정렬

분할 정복 알고리즘을 이용한 최선과 최악이 같은 안정적인 정렬 알고리즘
O(n log n) 시간복잡도를 가진다.



퀵 정렬

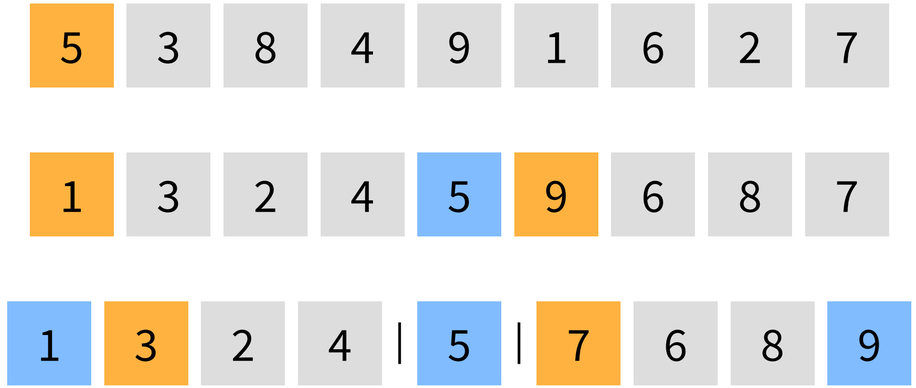
분할 정복 알고리즘을 이용한 매우 빠르지만 최악의 경우가 존재하는 불안정 정렬
O(n log n) 시간복잡도를 가진다.

퀵 정렬

분할 정복 알고리즘을 이용한 매우 빠르지만 최악의 경우가 존재하는 불안정 정렬
O(n log n) 시간복잡도를 가진다.

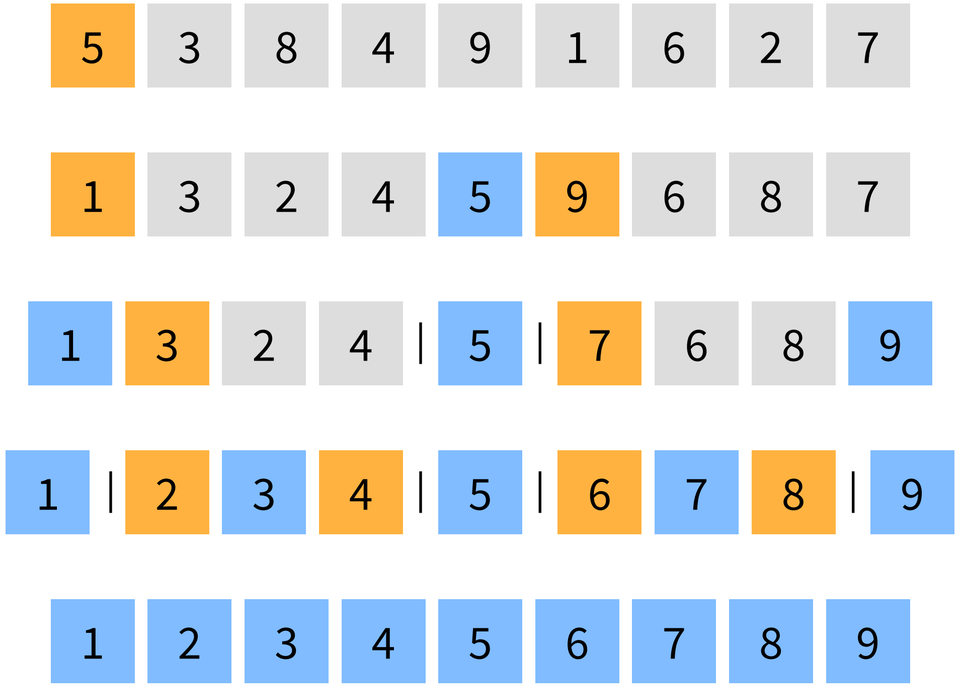
퀵 정렬

분할 정복 알고리즘을 이용한 매우 빠르지만 최악의 경우가 존재하는 불안정 정렬
O(n log n) 시간복잡도를 가진다.



퀵 정렬

분할 정복 알고리즘을 이용한 매우 빠르지만 최악의 경우가 존재하는 불안정 정렬
O(n log n) 시간복잡도를 가진다.



sort

```
const array = [5, 9, 10, 3, 8, 3, 2];
// 다음과 같이 그냥 정렬하면 ASCII 문자 순서로 정렬되어
// 우리가 원하는 숫자 크기대로 정렬되지 않는다.
array.sort();
console.log(array); // 10, 2, 3, 3, 5, 8, 9
// 10이 먼저 나오는 이유는 ASCII 문자 '1'이 '2'보다 작기 때문

array.sort((a, b) => a - b); // 오름차순 정렬
console.log(array); // 2, 3, 3, 5, 8, 9, 10

array.sort((a, b) => b - a); // 내림차순 정렬
console.log(array); // 10, 9, 8, 5, 3, 3, 2
```

programmers

정렬

코딩테스트 광탈방지 A to Z : JavaScript - 이선협 @kciter