

알고리즘 5

교재 참고 예제로 구성되어 있습니다
본 문서는 학습 참고용으로만 사용하시고 웨이나 타인에게 배포를 금합니다

버블정렬

선택정렬

삽입정렬

셸정렬

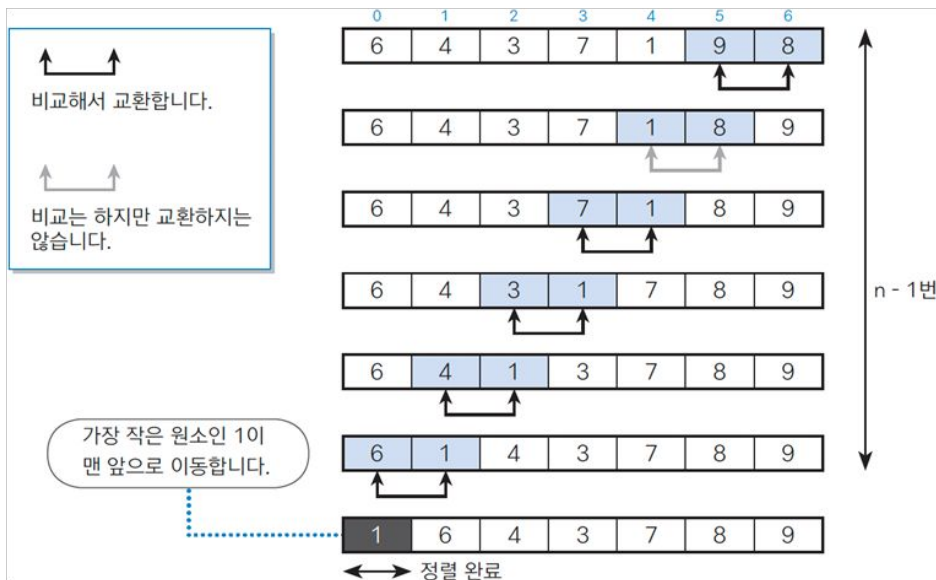
퀵정렬

병합정렬

버블정렬

버블 정렬 알고리즘은 서로 **인접한** 두 원소를 검사하여 정렬하는 알고리즘이다.

시간복잡도: $O(n^2)$



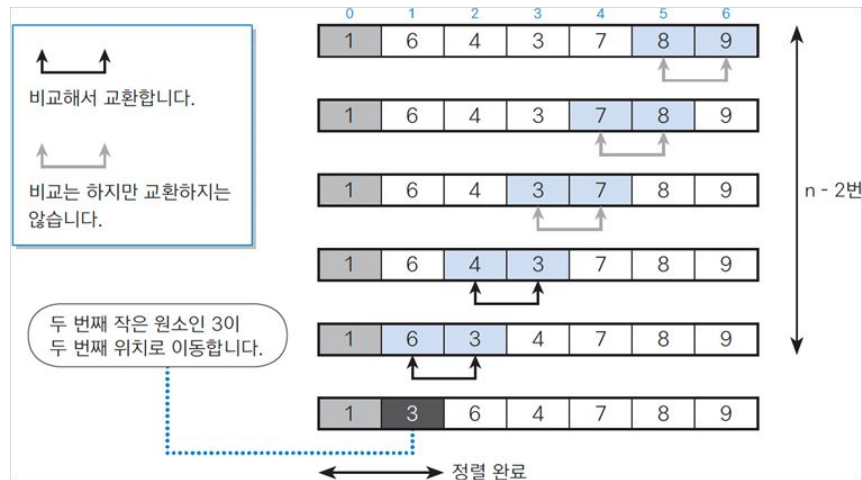
버블 정렬-Bubble Order

패스(pass): 비교 교환하는 과정

모든 정렬이 끝나려면 패스를 $n-1$ 번 수행

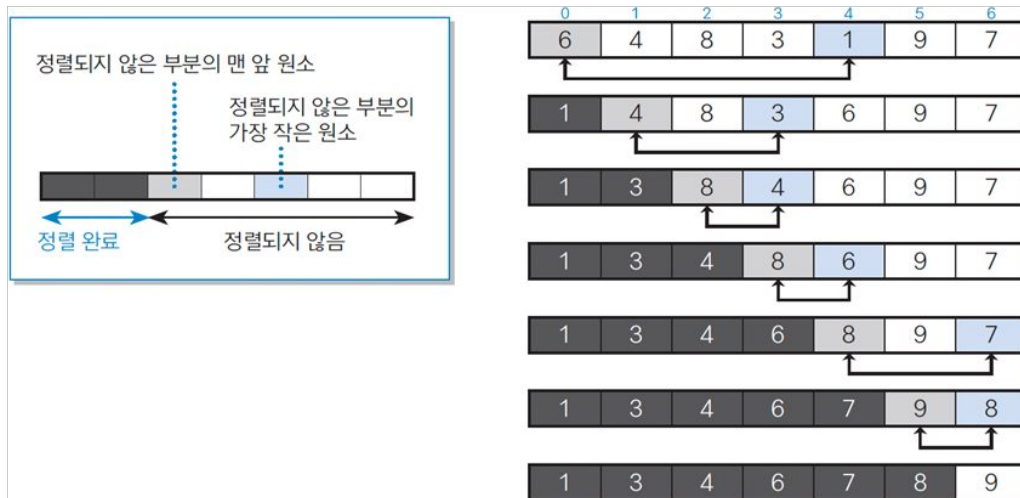
두 원소 $a[j-1]$ 과 $a[j]$ 의 값을 비교하여

앞쪽 값이 뒷쪽 값보다 크면 교환



선택 정렬 - Selection Sort

가장 작은 원소부터 선택해 알맞은 위치로 옮기는 작업을 반복하며 정렬하는 알고리즘



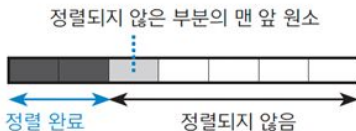
삽입정렬 - Insertion Sort

주목한 원소보다 더 앞쪽에서
알맞은 위치로 삽입하며 정렬하는
알고리즘

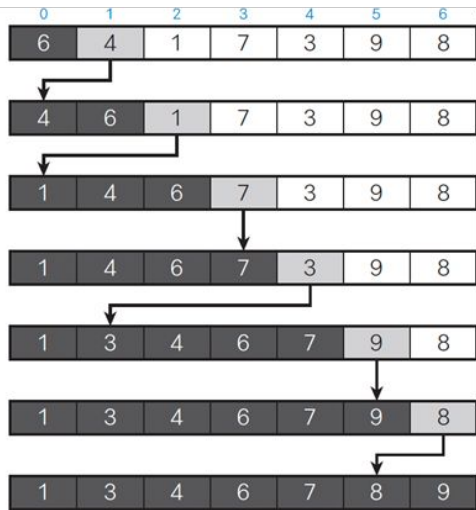
단순 선택 정렬과 비슷해 보이지만
값이 가장 작은 원소를 선택하지
않는다는 점이 다름

단순 정렬(버블, 선택, 삽입)
알고리즘의 시간 복잡도는 모두
 $O(n^2)$ 으로 프로그램의 효율이 좋지
않음

정렬되지 않은 부분의 맨 앞 원소를 정렬
된 부분의 알맞은 위치에 삽입하는 작업을
 $n - 1$ 번 반복합니다.



정렬 완료 $a[0], a[1], \dots, a[i - 1]$
정렬되지 않음 $a[i], a[i + 1], \dots, a[n - 1]$



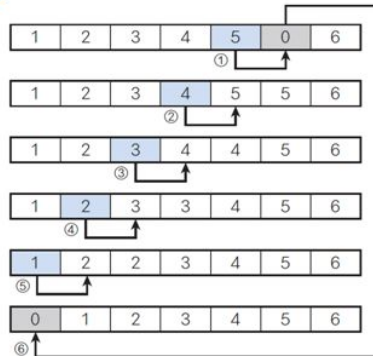
단순 삽입 정렬의 문제

▪ 단순 삽입 정렬의 한계

1	2	3	4	5	0	6
---	---	---	---	---	---	---



- ①~⑤ ... 0보다 작은 원소를 만날 때까지 이웃한 왼쪽 원소를 하나씩 대입하는 작업을 반복합니다.
⑥ ... 멈춘 위치에 0을 대입합니다.



- 여섯 번째 원소인 0을 삽입 정렬하려면 총 6번에 걸쳐 원소를 이동(대입)해야 함

▪ 단순 삽입 정렬의 특징

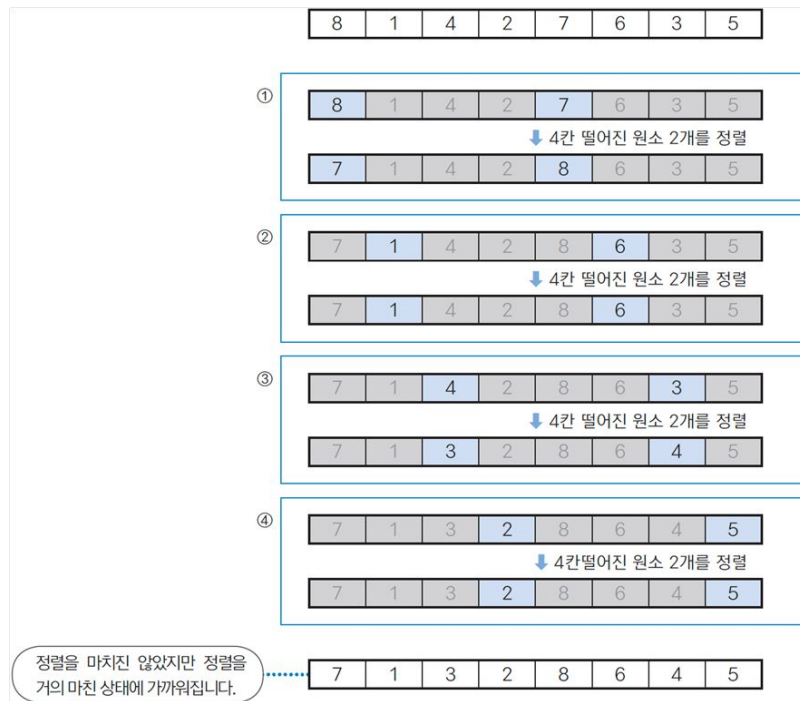
- **장점:** 이미 정렬을 마쳤거나 정렬이 거의 끝나가는 상태에서는 속도가 아주 빠름
- **단점:** 삽입할 위치가 멀리 떨어져 있으면 이동 횟수가 많아짐

셸 정렬-Shell Sort

단순 삽입 정렬의 장점을 살리면서 단점을 보완

먼저 정렬할 배열의 원소를 그룹으로 나눠 각
그룹별로 정렬을 수행한 뒤 정렬된 그룹을 합치는
작업을 반복하여 원소의 이동 횟수를 줄이는 방법

시간 복잡도는 $O(n^{1.25})$ 로 단순 정렬의 시간
복잡도인 $O(n^2)$ 보다 매우 빠르지만, 셸 정렬
알고리즘은 이웃하지 않고 떨어져 있는 원소를
서로 교환하므로 안정적인 정렬이 아님



셸 정렬-Shell Sort



퀵 정렬-Quick Sort

그룹을 나누는 기준을 정해 정렬하는

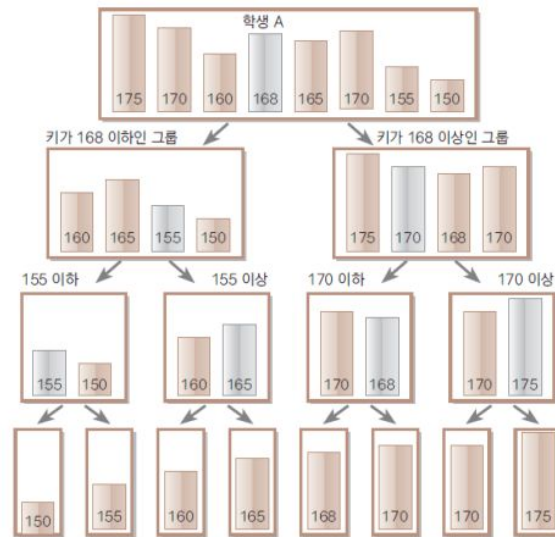
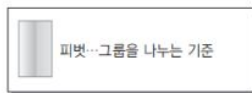
방법으로 일반적으로 사용되는 빠른 정렬

알고리즘이다

배열을 조금씩 나누어 보다 작은 문제를

푸는 과정을 반복하므로

시간 복잡도는 $O(n \log n)$



퀵 정렬-Quick Sort

배열을 두 그룹으로 나누기

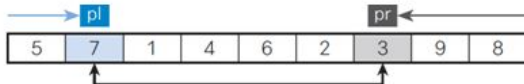
배열을 두 그룹으로 나누는 순서



- x: 피벗
- pl: 왼쪽 끝 원소의 인덱스(왼쪽 커서)
- pr: 오른쪽 끝 원소의 인덱스(오른쪽 커서)

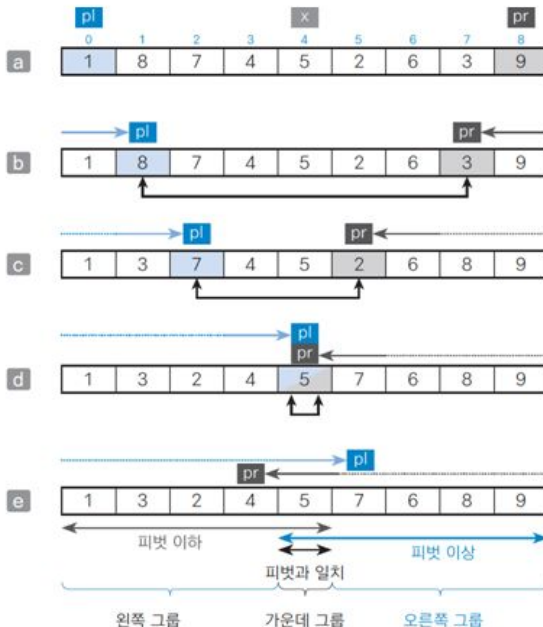
- 피벗 이하인 원소를 배열 왼쪽으로, 피벗 이상인 원소를 배열 오른쪽으로 이동시키는 것이 목표

- $a[pl] \geq x$ 인 원소를 찾을 때까지 pl을 오른쪽 방향으로 스캔
- $a[pr] \leq x$ 인 원소를 찾을 때까지 pr을 왼쪽 방향으로 스캔



- pl과 pr가 위치하는 원소 $a[pl]$ 과 $a[pr]$ 의 값을 교환

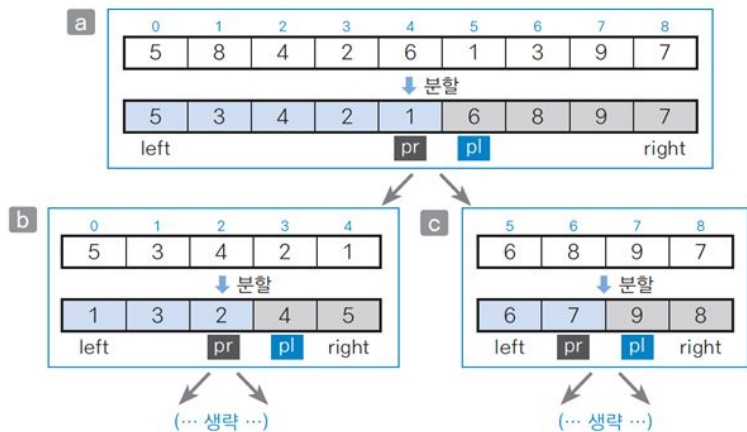
피벗과 일치하는 그룹이 생성된 예



퀵 정렬-Quick Sort

■ 퀵 정렬 알고리즘

■ 퀵 정렬의 배열 나누기



■ 원소 수가 2개 이상인 그룹을 반복해서 나누기

- pr가 a[0]보다 오른쪽에 위치하면(left < pr) 왼쪽 그룹 나누기
- pl이 a[8]보다 왼쪽에 위치하면(pl < right) 오른쪽 그룹 나누기

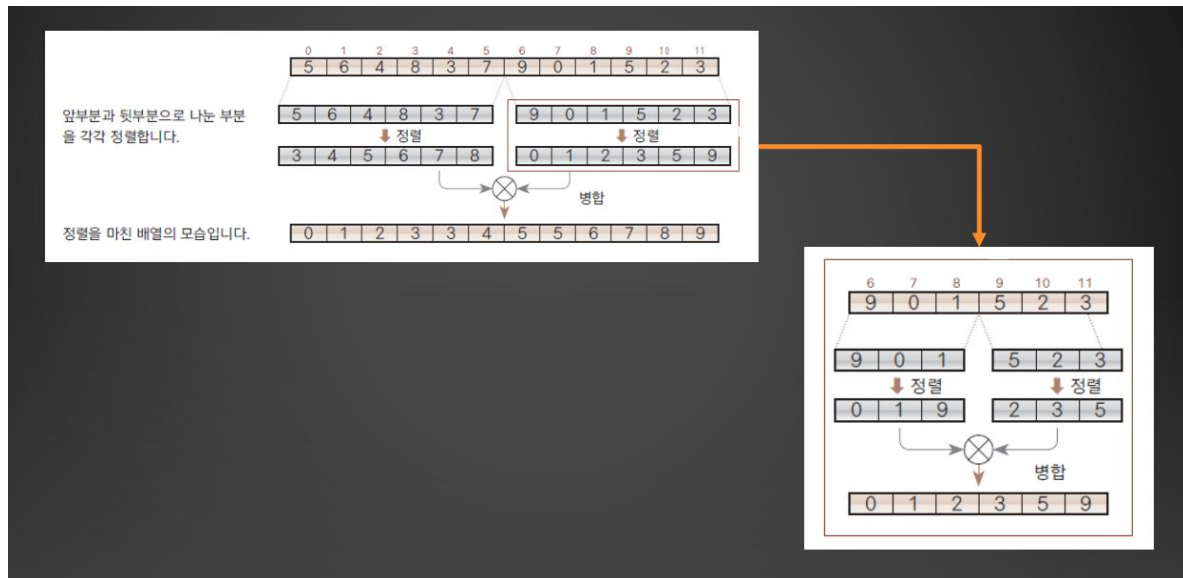
■ 퀵 정렬은 8퀸 문제와 같은 분할 정복 알고리즘이므로 재귀 호출을 사용하여 구현 가능

■ 퀵 정렬은 서로 이웃하지 않는 원소를 교환하므로 안정적이지 않은 알고리즘

병합 정렬-Merge Sort

배열을 앞부분과 뒷부분 두
그룹으로 나누어 각각 정렬한
후 병합하는 작업을 반복하는
알고리즘

시간복잡도는 $O(n \log n)$



병합 정렬-Merge Sort

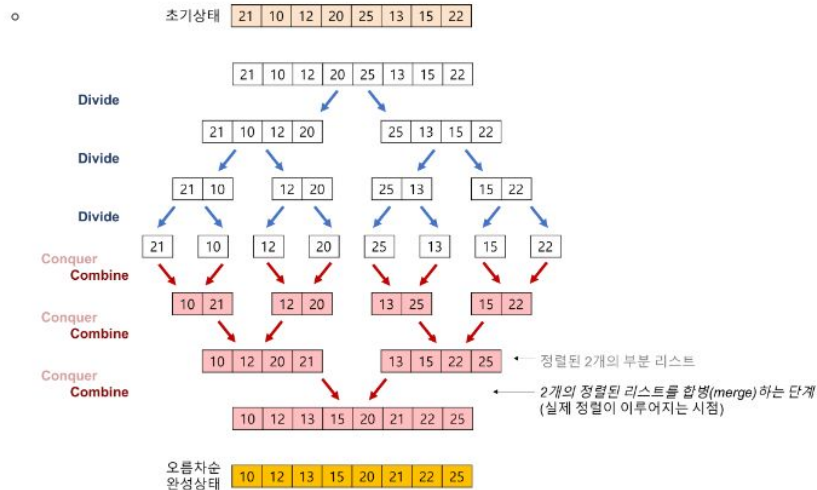
하나의 배열을 두 개의 균등한 크기로 분할하고 분할된 부분 배열을 정렬한 다음, 두 개의 정렬된 부분 배열들을 합하여 전체가 정렬된 배열이 되게 하는 방법이다.

병합 정렬은 다음의 단계들로 이루어진다.

[1] 분할(Divide): 입력 배열을 같은 크기의 2개의 부분 배열로 분할한다.

[2] 정복(Conquer): 부분 배열을 정렬한다. 부분 배열의 크기가 충분히 작지 않으면 순환 호출을 이용하여 다시 분할 정복 방법을 적용한다.

[3] 결합(Combine): 정렬된 부분 배열들을 하나의 배열에 합병한다.



병합 과정

- 초기상태

21	10	12	20	25	13	15	22
----	----	----	----	----	----	----	----

