정렬

https://youtu.be/Hxtxmi9icHE

Bubble Sort

Insertion Sort

Selection Sort

Merge Sort

Quick Sort

정렬 알고리즘 개요

보통 시간 복잡도에 따라 알고리즘을 분류

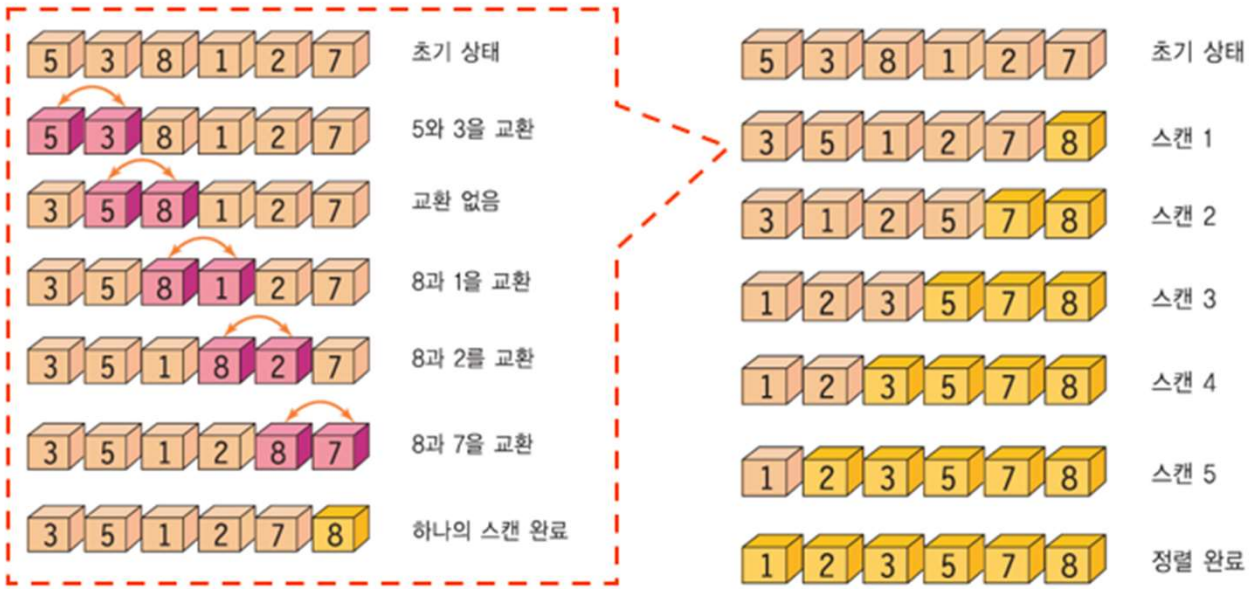
복잡도	설명	대표 알고리즘
O(1)	자료 크기 무관 항상 같은 속도	해시 함수
O($\log_2 n$)	$\log_2 n$번만큼 수행 시간 가짐	이진 탐색
O(n)	입력 자료를 하나씩 처리	순차 탐색
O($n\log_2 n$)	$n\log_2 n$번만큼 수행 시간 가짐	퀵정렬, 합병정렬, 힙정렬
O(n^2)	루프 구조가 2중인 경우 n크기가 작으면$n\log_2 n$보다 빠를 수 있음	버블정렬, 삽입정렬,선택정렬

Bubble Sort

Bubble Sort: 인접한 2개의 레코드 키값을 비교하여 위치 교환.

→한 PASS를 수행할 때 가장 큰 값이 맨 뒤로 이동하기 때문에,

‘요소의 개수-1’번 수행하면 모든 숫자 정렬 가능.



비교:O(n^2)

→ 최상, 평균, 최악 모두 일정

이동

→ 역순: 3x 비교 횟수

정렬된 경우: 0

Insertion Sort

Insertion Sort: n번째 키를 앞의(n-1)개 키와 비교하여 알맞은 위치에 삽입하는 과정을 반복

자료 배열의 모든 요소를 앞에서부터 차례대로 이미 정렬된 배열 부분과 비교.

최선의 경우O(n) → 이미 정렬되어 있는 경우

비교: n-1

최악의 경우O(n^2) → 역순으로 정렬되어 있는 경우

비교:O(n^2), 이동:O(n^2)

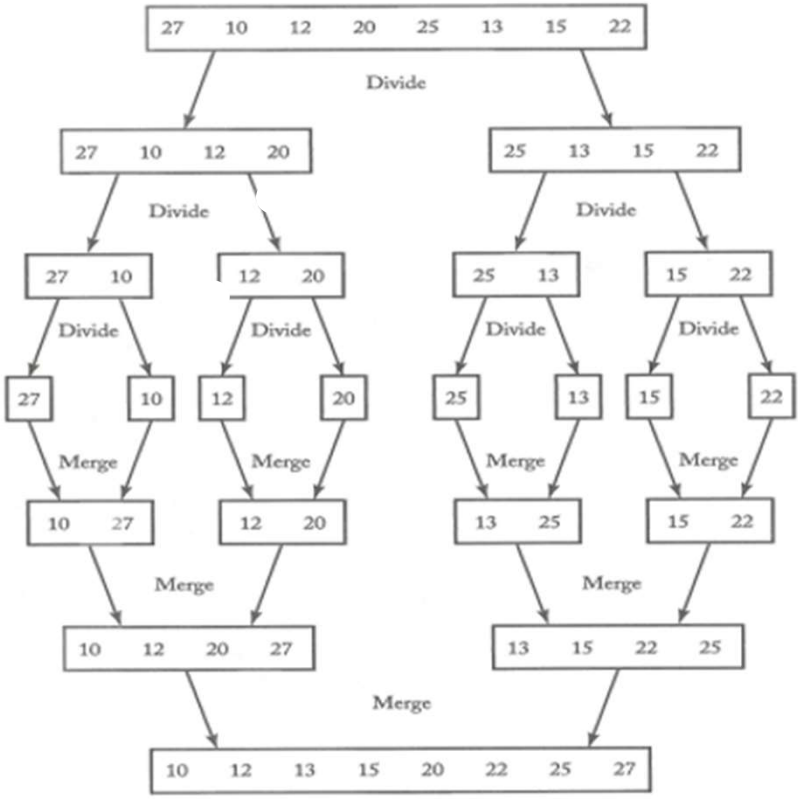
Selection Sort

Selection Sort: 정렬되지 않은 데이터들에 대해 가장 작은 데이터를 찾아 정렬되지 않은 부분의 가장 앞의 데이터와 교환해나가는 방법.

비교횟수: O(n^2), 이동횟수:3(n-1)

Merge Sort

Merge Sort:전체 원소를 하나의 단위로 분할 후 다시 합쳐 정렬



비교횟수: 크기 n의 리스트를 정확인 균등 분배

→ $\log n$개의 패스

각 패스에서 모든 레코드의 n개를 비교하므로

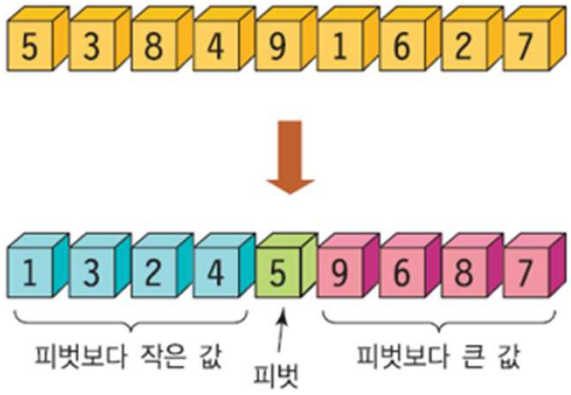
n번의 비교연산

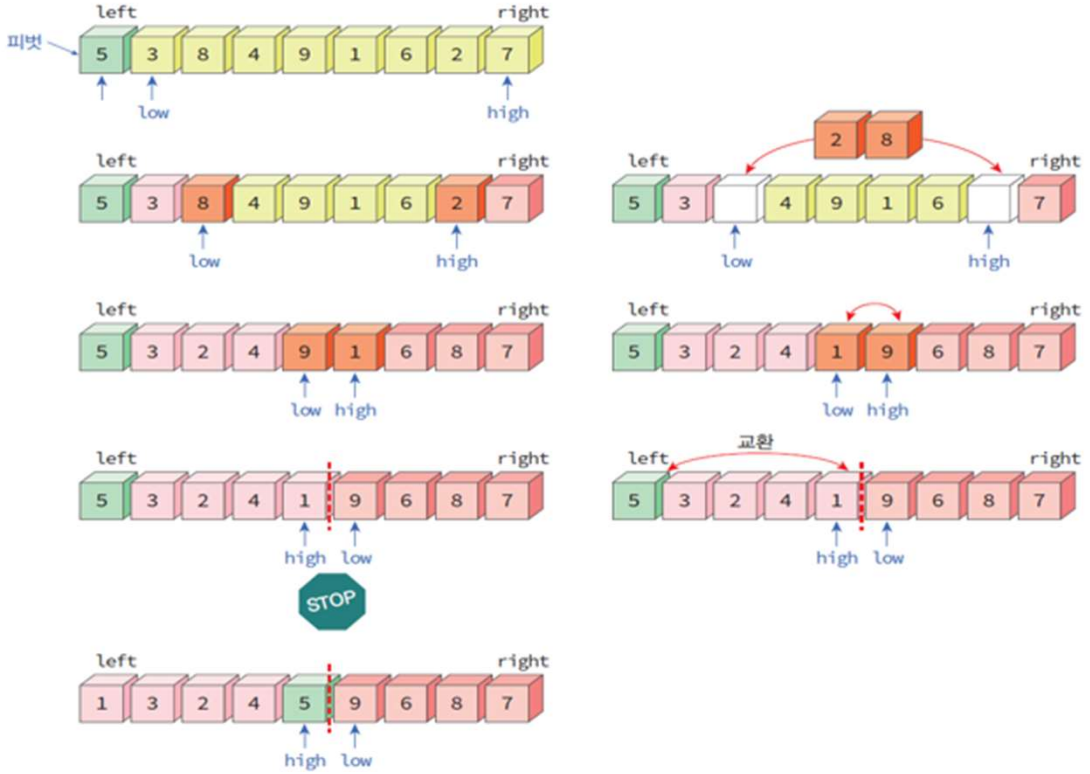
이동횟수:레코드 이동이 각 패스에서 2n번 발생

→ 전체 레코드 이동은 2n* $\log n$ 번 발생

Quick Sort

Quick Sort: 피벗을 두고 피벗의 왼쪽에는 피벗보다 작은 값, 오른쪽에는 큰 값을 두는 과정을 반복





Q & A