정렬 알고리즘

1. Selection Sort (선택 정렬)

- Time complexity : $O(n^2)$

⇒ 주어진 데이터들 중, 현재 위치에 맞는 데이터를 선택해 위치를 교환

ex) 5 4 3 1 2

① 5를 선택, 나머지 중 가장 작은 데이터와 교환 ⇒ 1 4 3 5 2

② 4를 선택, 〃 ⇒ 1 2 3 5 4

⋮

최종적으로 12345로 정렬된다!

2. Insertion Sort (삽입정렬)

- Time complexity : $O(n^2)$ (worst case), $O(n)$ (best case)

⇒ 맨 왼쪽을 기준으로 한칸씩 오른쪽으로 비교 값을 옮기는데,
이 기준값 보다 앞에 있는 모든 배열들과 비교하며 기준값이 비교한 값보다 작다면, 비교값들을 크거나 같을 때까지 오른쪽으로 당긴다.
(이때, 첫번째 기준 값일때는 제외하고 앞서 말한 앞에 있는 모든 배열들은 몇 차례 삽입정렬에 의해 sorting 된 것이다)

ex)

6	4	3	1	2

① 기준값: 6

[6] 4 3 1 2 ⇒ 비교할 배열이 없음 (skip!)

② 기준값: 4, 비교값(배열): [6]

(1) [6] [4] 3 1 2 ⇒ 4와 6 비교

(2) 4 6 3 1 2

③ 기준값: 3, 비교값(배열): [4, 6]

(1) [4 6] [3] 1 2

(2) [3] ⇐
[4 ⇒ 6] 1 2 ⇒ 6과 3 비교

(3) [3 4 6] 1 2 ⇒ 4와 3 비교 + 더 비교할 수 있는 값이 없으므로 값 넣기

④ 기준값: 1, 비교값(배열): [3, 4, 6]

(1) [3 4 6] [1] 2

(2) [1]
[3 4] [6] 2

(3) [1]
[3] [4 6] 2

(4) [1] [3 4 6] 2

⋮

최종적으로, [1 2 3 4 6] 배열 완성!

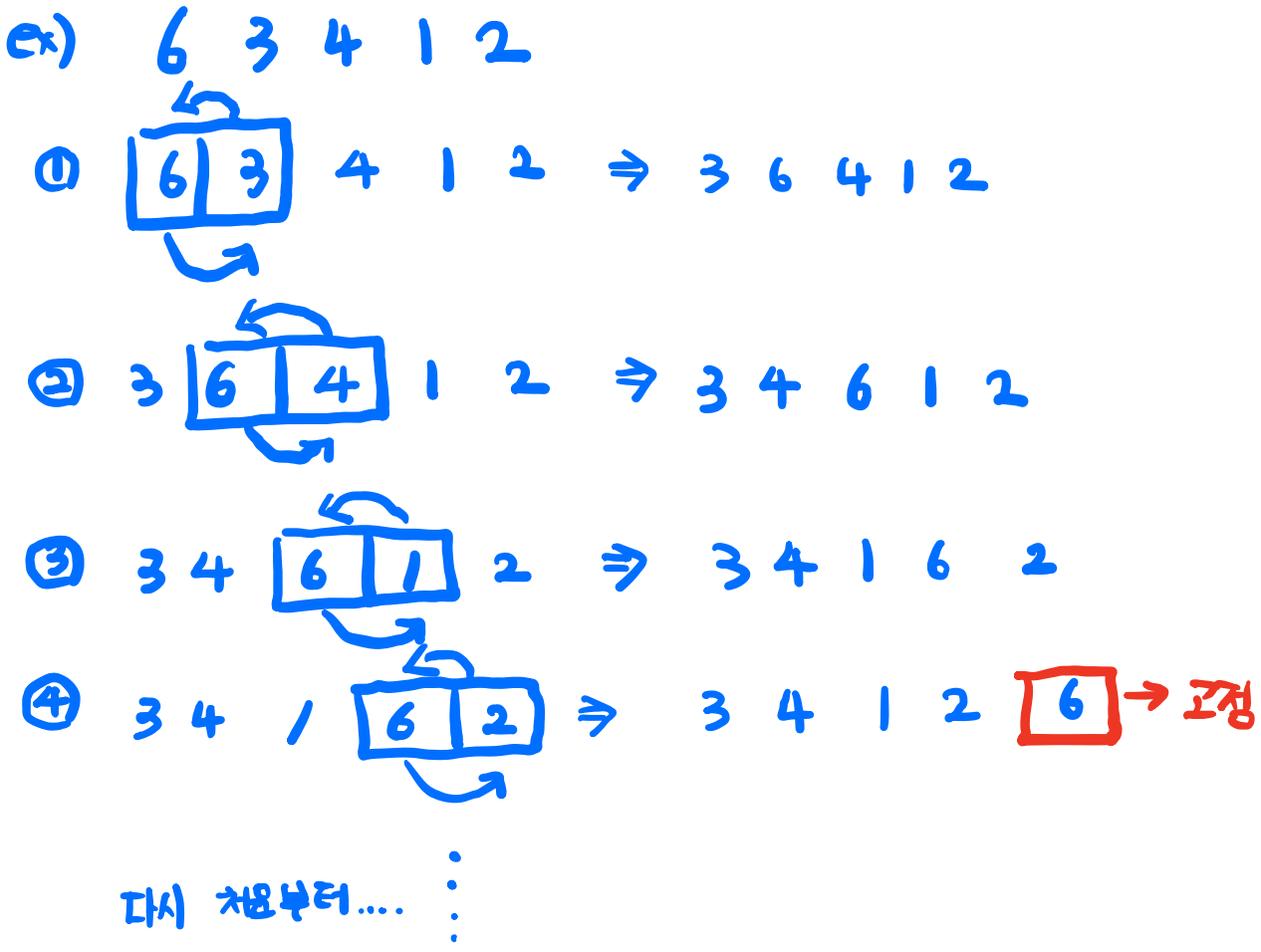
⇒ 정렬되어있는 배열 일 수록, 효율이 ↑

3. Bubble Sort (버블정렬)

- Time Complexity : $O(n^2)$ (조건에 맞을 때)

⇒ 배열에서 두 인접한 원소를 검사해서 자리를 바꾸는 정렬 방법이다.

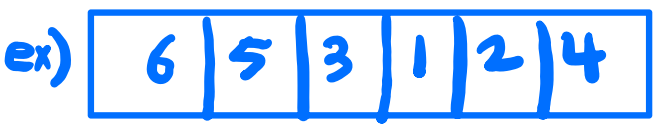
(제일 큰 값을 맨 뒤로 차곡 차곡 쌓는다고 생각하면 편하다)

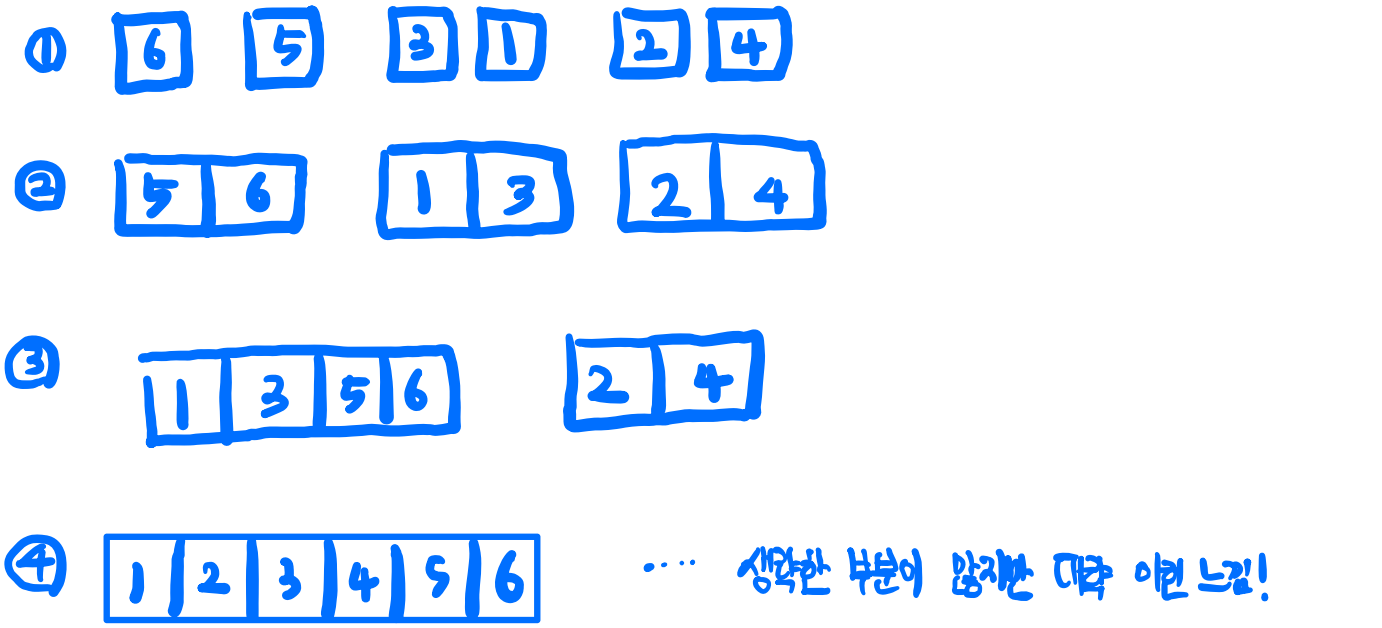


4. Merge Sort (병합정렬)

- Time Complexity : $O(n \log n)$

⇒ 주어진 배열이 1이 될때까지 쪼갠 후, 작은 단위들부터 정렬하며 정렬된 단위들을 계속 병합해가며 정렬한다.

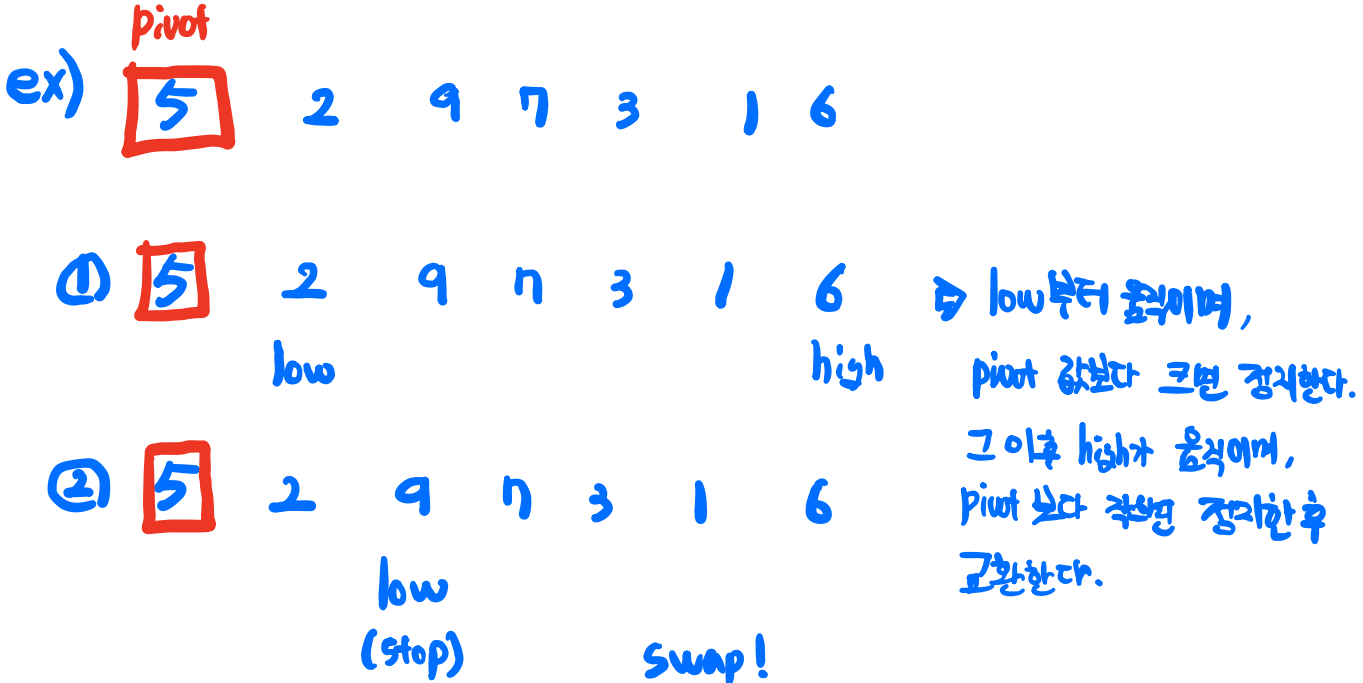




5. Quick Sort (퀵 정렬)

- Time Complexity : $O(n \log n)$ (Best), $O(n^2)$ (worst)

⇒ 연속적인 분할에 의한 정렬 방법이다. 처음 배열의 축 (pivot)을 기준으로 이 값 보다 작으면 왼쪽으로, 큰 값은 오른쪽으로 위치시키고 low와 high가 교차될 때까지 한다. 마지막으로 high값과 pivot값을 교환한다. 이후 high와 low가 각자 가르키던 값들이 pivot 이 되어서 정렬을 계속한다.



③ [5] 2 4 7 3 6

low (stop) high (stop)

④ [5] 2 1 7 3 9 6

low high

[stop]

swap!

⑤ [5] 2 1 7 3 9 6

low high

[stop]

⑥ [5] 2 1 3 7 9 6

high low

[stop] [stop]

⇒ 중단 됐음에도 low와 high의 위치가 변경 됐으므로 교환하지 않고 정렬을 끝낸다.

swap

⑦ [5] 2 1 (3) 7 9 6 ⇒ 1차 정렬 후, pivot 값과 high 값을 바꿔준다.

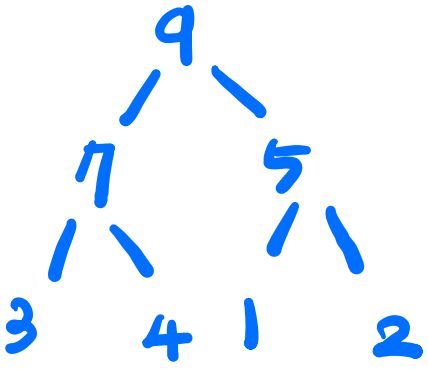
⑧ 3 2 1 [5] 7 9 6 ⇒ 5를 제외하고 left의 pivot은 3 right pivot은 7 이제부터 분할 정렬을 진행한다!

6. Heap Sort (힙 정렬)

• Time Complexity: $O(n \log n)$ //

⇒ 최대 & 최소 힙 트리를 구성해서 정렬하는 방법 (내림차순은 - MAX / 오름차순은 - MIN)

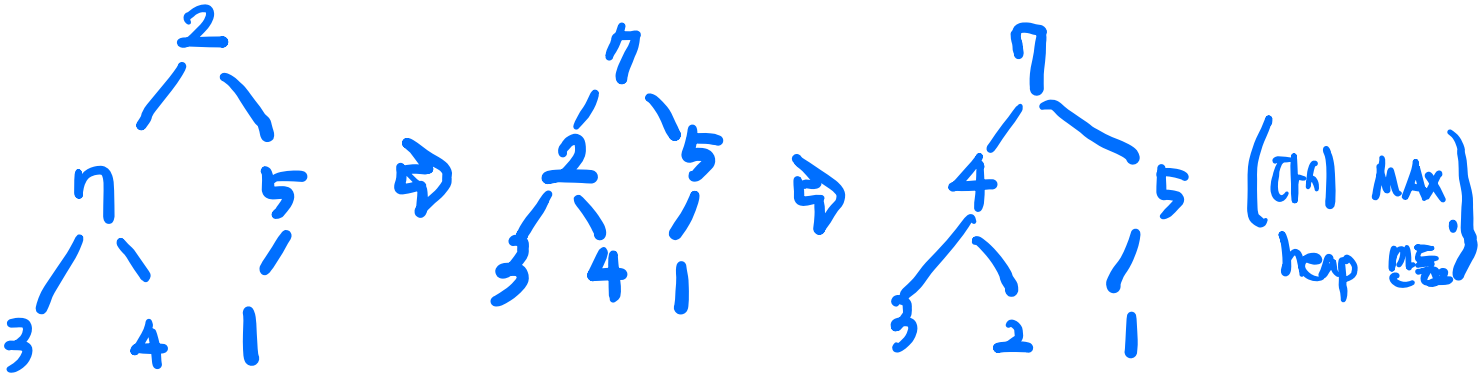
ex) MAX heap 아닌 경우



9	7	5	3	4	1	2

① 9 ↔ 2 (제일 마지막 node와 root node 바꿈)

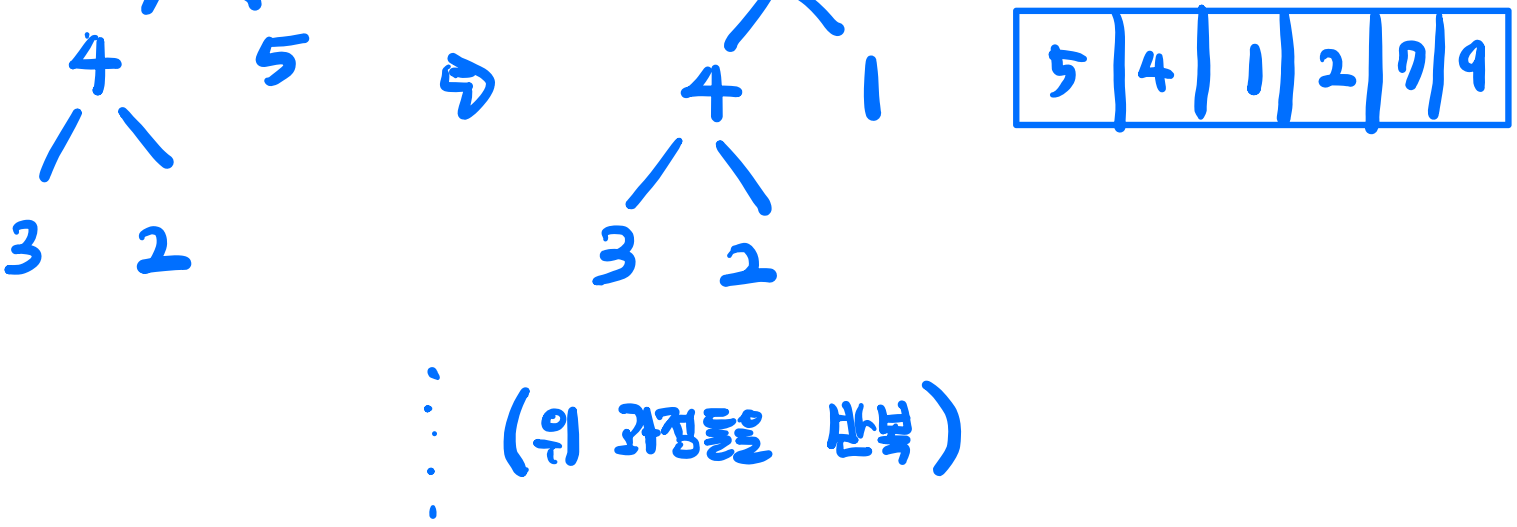
삭제 (pop) 라고 생각하면 편함!



7	4	5	3	2	1	9

고정

② 7 ↔ 1



(위 과정들을 반복)

정렬 종류	시간 복잡도			공간 복잡도
	평균 (Average)	최선(Best)	최악(Worst)	
선택 정렬	O(n2)	O(n2)	O(n2)	O(n2)
버블 정렬	O(n2)	O(n2)	O(n2)	O(n)
삽입 정렬	O(n2)	O(n)	O(n2)	O(n2)
합병 정렬	O(n×log n)	O(n×log n)	O(n×log n)	O(n×log n)
퀵 정렬	O(n×log n)	O(n×log n)	O(n2)	O(n×log n)
힙 정렬	O(n×log n)	O(n×log n)	O(n×log n)	O(n×log n)