

# 제9장 정렬

## 1. 개요

정렬(sort)은 자료를 **크기 순**으로 재배열하는 것이다.

- 오름차순정렬(ascending) : 작은 순서 → 큰 순서(비내림차순)
- 내림차순정렬(descending) : 큰 순서 → 작은 순서(비오름차순)

먼저, 정렬 알고리즘 종류별로 일반적인 특징을 정리하면 다음과 같다.

정렬방법	수행시간	최악의 경우	기억공간	안정/불안정	제자리정렬
선택정렬	$O(n^2)$	$O(n^2)$	n	불안정	○
거품정렬	$O(n^2)$	$O(n^2)$	n	안정적	○
삽입정렬	$O(n^2)$	$O(n^2)$	n	안정적	○
셸정렬	$O(n^2)$	$O(n^2)$	n	불안정	○
힙정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	n	불안정	○
합병정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	2n	안정적	×
퀵정렬	$O(n \log_2 n)$	$O(n^2)$	n+stack	불안정	○ / ×
기수정렬	$O(k(n+q))$	$O(k(n+q))$	(n+1)q	안정적	×

### ① 정렬 장소에 따른 구분

- 내부정렬 : 정렬 대상 모든 자료를 **주기억장소**에 적재시켜서 정렬하는 방법
- 외부정렬 : 자료의 양이 너무 많아서 **보조기억장소**를 이용하여 정렬을 수행

### ② 정렬 알고리즘 선택에 영향을 미치는 요인

- 컴퓨터 시스템의 특성 및 정렬 대상이 되는 자료의 양
- 정렬 대상 자료의 초기 배열 상태 및 키 값의 분포 상태
- 소요공간 및 수행시간(**알고리즘 복잡도**)



## (2) 제자리정렬(in-place sort)

- 제자리정렬은 정렬 과정에서 추가 기억장소를 조금만 더 사용하는 알고리즘이다.
  - 제자리정렬은 정렬 대상 자료 이외의 추가 기억장소가 상수 개를 넘지 않는 알고리즘이다.
  - 제자리정렬은 정렬 대상 자료에 비해 무시할 정도의 기억장소를 더 사용하는 알고리즘이다.
- 제자리정렬 : 선택정렬, 버블정렬, 삽입정렬, 셸정렬, 힙정렬

### ◆ 퀵정렬

- 퀵정렬은 제자리정렬로 분류할 수도 있고 아닐 수도 있다.
- 퀵정렬은 재귀 알고리즘으로 스택을 사용하는데, 스택의 깊이 공간복잡도는 상수가 아니다.
- 하지만, 퀵정렬은 실제 사용에서는 적은 메모리만을 더 사용하므로 제자리정렬로 분류된다.

### ◆ 합병정렬

- 기본적으로, 합병정렬은 제자리 정렬(in-place sort)이 아니다.
- 하지만, 합병정렬은 제자리 정렬로 구현할 수 있다.
- 합병정렬의 종류는 많다.
- 합병정렬은 반복 또는 순환으로 구현할 수 있다.

반복 합병정렬	<ul style="list-style-type: none"> <li>• 제자리 정렬이 아니다.</li> <li>• 합병정렬을 수행하기 위한 임시 배열이 추가로 필요하다.</li> <li>• 레코드 크기가 큰 경우에 이동 횟수가 많으므로 시간 낭비를 초래한다.</li> </ul>
개량된 순환 합병정렬	<ul style="list-style-type: none"> <li>• 제자리 정렬로 구현할 수 있다.</li> <li>• 개량된 순환 합병정렬은 연결리스트를 이용하여 구현한다.</li> <li>• 개량된 순환 합병정렬에서 추가 메모리는 n개의 포인터 필드만 필요로 한다.</li> <li>• 정렬 과정에서 포인터만 변경하면 되므로 물리적인 데이터 이동은 없다.</li> <li>• 크기가 큰 레코드를 정렬할 때, 연결리스트를 사용하면 합병정렬은 효율적이다.</li> </ul>

- 위에 설명된 내용에 합병정렬 부분에 구체적으로 설명되어 있다.

**기출문제 분석**

1. '어떤 정렬 알고리즘을 선택할 것인가?' 고려할 사항으로 거리가 먼 것은? [2004년 기술고시]

- ① 초기 데이터의 배열 상태
- ② 키 값들의 분포상태
- ③ 소요공간 및 작업시간
- ④ 레코드 내에서 키 필드 위치

☞ 정렬 알고리즘 선택에 영향을 미치는 요인

---

- 레코드 내에서 키 필드 위치는 무관하다.

// 정렬 알고리즘 선택에 영향을 미치는 요인

- 컴퓨터 시스템의 특성 및 정렬 대상이 되는 자료의 양
  - 정렬 대상 자료의 초기 배열 상태 및 키 값의 분포 상태
  - 소요공간 및 수행시간(알고리즘 복잡도)
- 

정답 : ④

2. 주어진 자료에 맞는 정렬(sort) 알고리즘을 선택할 때 고려해야 할 사항으로 가장 옳지 않은 것은? [2022년 서울 7급]

- ① 필요한 작업공간 및 소요시간
- ② 정렬할 자료의 양
- ③ 정렬에 필요한 메모리(memory)의 크기
- ④ 추가되는 데이터의 배열 상태

☞ 정렬 알고리즘 선택에 영향을 미치는 요인

---

- 컴퓨터 시스템의 특성 및 정렬 대상이 되는 자료의 양
  - 정렬 대상 자료의 초기 배열 상태 및 키 값의 분포 상태
  - 정렬에 필요한 메모리(memory)의 크기
  - 작업공간 및 소요시간(알고리즘 복잡도)
- 

정답 : ④

3. 비교가 아닌 분배에 의한 정렬(sorting by distribution) 방식은? [2011년 국가 7급]

- ① 기수정렬                      ② 버블정렬
- ③ 퀵정렬                        ④ 히프정렬

☞ 기수정렬

- 
- 기수정렬은 여러 개의 키를 가지는 레코드들을 정렬하는 분야에 적용된다.
  - 각 키 단위를 기준으로 저장통(bin, bucket)에 분배하는 방식으로 정렬을 수행한다.
- 

정답 : ①

4. 최악 시간복잡도(worst-case time complexity)가  $O(n \log n)$ 인 정렬 방식만을 모은 것은?  
(단,  $n$ 은 데이터의 개수이다) [2014년 국가 7급] [2019년 군무 7급 유형]

- ① 합병(merge)정렬, 힙(heap)정렬                      ② 삽입(insertion)정렬, 버블(bubble)정렬
- ③ 선택(selection)정렬, 퀵(quick)정렬                ④ 퀵(quick)정렬, 힙(heap)정렬

☞ 최악 시간복잡도

- 
- 합병(merge)정렬, 힙(heap)정렬 :  $O(n \log n)$
  - 삽입(insertion)정렬, 버블(bubble)정렬, 선택(selection)정렬, 퀵(quick)정렬 :  $O(n^2)$
- 

정답 : ①

5. 키 값에 대한 연산을 비교와 상호교환으로 한정된 내부 정렬 알고리즘에서 가능한 최적의 연산 시간복잡도? [2003년 국가 7급]

- ①  $O(n)$                               ②  $O(n \log_2 n)$
- ③  $O(n^2)$                             ④  $O(n^2 \log_2 n)$

☞ 비교와 상호 교환 방식의 내부정렬

- 
- 비교와 상호교환으로 한정된 내부정렬 알고리즘의 복잡도는  $O(n \log_2 n)$ 이다.
  - 비교와 상호교환 정렬 알고리즘의 복잡도는  $O(n \log_2 n)$ 보다 더 우수할 수 없다는 사실이 학자들에 의해 검증되었다.
  - 예 : 버블정렬, 퀵정렬, 힙정렬 등의 복잡도는  $O(n \log_2 n)$ 보다 더 우수할 수 없다.
- 

정답 : ②

6. 정렬 알고리즘에 대한 설명 중 가장 적절하지 않은 것은? [2021년 군무원 7급]

- ① 삽입정렬(insertion sort), 선택정렬(selection sort), 퀵정렬(quick sort) 알고리즘은 점근분 석 시 최악수행시간이 같다.
- ② 삽입정렬, 힙정렬(heap sort), 퀵정렬은 제자리정렬(in-place sorting) 구현이 가능하다.
- ③ 합병정렬(merge sort), 힙정렬, 기수정렬(radix sort) 알고리즘은 원소 간 대소비교를 통 해 정렬하는 알고리즘 중 최적 알고리즘이다.
- ④ 선택정렬 알고리즘은 최악수행시간과 평균수행시간이 같다.

☞ 정렬 알고리즘

- 합병정렬(merge sort), 힙정렬, 기수정렬(radix sort) 알고리즘은 원소 간 대소비교를 통해 정렬하 는 알고리즘 중 최적 알고리즘이다.(×)
- 합병정렬은 정렬하는 알고리즘 중 최적 알고리즘이 아니다.
- 기수정렬은 원소 간 대소비교를 실시하지 않는다.

// 정렬 알고리즘 종류별로 일반적인 특징을 정리하면 다음과 같다.

정렬방법	수행시간	최악의 경우	기억공간	안정/불안정	제자리정렬
선택정렬	$O(n^2)$	$O(n^2)$	n	불안정	○
거품정렬	$O(n^2)$	$O(n^2)$	n	안정적	○
삽입정렬	$O(n^2)$	$O(n^2)$	n	안정적	○
셸정렬	$O(n^2)$	$O(n^2)$	n	불안정	○
힙정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	n	불안정	○
합병정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	2n	안정적	×
퀵정렬	$O(n \log_2 n)$	$O(n^2)$	n+stack	불안정	○ / ×
기수정렬	$O(k(n+q))$	$O(k(n+q))$	(n+1)q	안정적	×

// 제자리정렬(in-place sort)

- 제자리정렬은 정렬 과정에서 추가 기억장소를 조금만 더 사용하는 알고리즘이다.
- 제자리정렬은 정렬 대상 자료 이외의 추가 기억장소가 상수 개를 넘지 않는 알고리즘이다.
- 제자리정렬은 정렬 대상 자료에 비해 무시할 정도의 기억장소를 더 사용하는 알고리즘이다.

// 퀵정렬

- 퀵정렬은 제자리정렬로 분류할 수도 있고 아닐 수도 있다.
- 퀵정렬은 재귀 알고리즘으로 스택을 사용하는데, 스택의 깊이 공간복잡도는 상수가 아니다.
- 하지만, 퀵정렬은 실제 사용에서는 적은 메모리만을 더 사용하므로 제자리정렬로 분류된다.

7. 다음의 정렬 방법 중 분할정복(divide-and-conquer)의 개념을 이용한 정렬 방법만 골라 놓은 것으로 가장 적절한 것은? [2021년 군무원 7급]

- |          |                  |          |
|----------|------------------|----------|
| (가) 삽입정렬 | (나) 퀵정렬          | (다) 힙정렬  |
| (라) 병합정렬 | (마) 기수(radix) 정렬 | (바) 버블정렬 |

- ① (가), (바)      ② (나), (라)  
 ③ (다), (라)      ④ (라), (마)

☞ 정렬 방법 중 분할정복

// 퀵정렬 단계

분할 (divide)	<ul style="list-style-type: none"> <li>정렬 대상 리스트(배열)를 피벗을 기준으로 2개의 부분 리스트로 분할한다.</li> <li>피벗을 중심으로 왼쪽 : 피벗보다 작은 요소들</li> <li>피벗을 중심으로 오른쪽 : 피벗보다 큰 요소들</li> <li>분할된 2개의 부분 리스트는 비균등한 구조가 된다.</li> </ul>
정복 (conquer)	<ul style="list-style-type: none"> <li>분할된 각 부분 리스트를 정렬한다.</li> <li>재귀호출을 이용하여 다시 분할 정복 방법을 적용한다.</li> <li>부분 리스트 크기가 충분히 작아질 때까지 재귀호출을 적용한다.</li> <li>부분 리스트 크기가 0이나 1이 될 때까지 재귀호출 된다.</li> </ul>
결합 (combine)	<ul style="list-style-type: none"> <li>정렬된 부분 리스트를 하나의 리스트(배열)에 합병한다.</li> <li>재귀호출이 한번 진행될 때마다 최소 하나의 원소(피벗) 위치가 결정된다.</li> <li>해서, 퀵정렬 알고리즘은 반드시 종료된다는 것을 보장할 수 있다.</li> </ul>

- 합병정렬은 균등하게 분할된다.
- 퀵정렬은 합병정렬과 달리 리스트를 비균등하게 분할된다.

정답 : ②

8. 정렬에서 키 값을 비교하는 방법이 아닌 것은? [2002년 서울 7급]

- ① Merge Sort      ② Bubble Sort      ③ Bucket Sort  
 ④ Quick Sort      ⑤ Heap sort

☞ Bucket Sort

- Bucket Sort는 분배법이다. 큐에 자료를 분배하는 원리로 정렬을 수행한다.

정답 : ③

9. 다음의 정렬 알고리즘 중 안정정렬 알고리즘으로 가장 옳지 않은 것은? [2022년 군무원 7급]

- ① 버블(bubble)정렬
- ② 선택(selection)정렬
- ③ 삽입(insertion)정렬
- ④ 합병(merge)정렬

☞ 정렬 알고리즘

정렬방법	수행시간	최악의 경우	기억공간	안정/불안정	제자리정렬
선택정렬	$O(n^2)$	$O(n^2)$	n	불안정	○
거품정렬	$O(n^2)$	$O(n^2)$	n	안정적	○
삽입정렬	$O(n^2)$	$O(n^2)$	n	안정적	○
셸정렬	$O(n^2)$	$O(n^2)$	n	불안정	○
힙정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	n	불안정	○
합병정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	2n	안정적	×
퀵정렬	$O(n \log_2 n)$	$O(n^2)$	n+stack	불안정	○ / ×
기수정렬	$O(k(n+q))$	$O(k(n+q))$	(n+1)q	안정적	×

선택정렬에서 불안정한 경우	초기자료 : 5 <sub>1</sub> 9 5 <sub>2</sub> 8 3
	단계 1 : 3 9 5 <sub>2</sub> 8 5 <sub>1</sub> → 5 <sub>1</sub> 와 5 <sub>2</sub> 의 위치가 바뀜

• 키 값이 같은 레코드들의 정렬 순서가 보존되는 정렬을 안정적인 정렬이라 한다.

정답 : ②

10. 정렬에서 재귀적인 방법을 사용하는 정렬 알고리즘으로 짝지어진 것은? [2000년 서울 7급]

- ① Selection Sort, Bubble Sort
- ② Insertion Sort, Quick Sort
- ③ Quick Sort, Heap Sort
- ④ Quick Sort, Merge Sort
- ⑤ Heap Sort, Merge Sort

☞ 재귀적 정렬 알고리즘

• Quick Sort는 재귀적인 알고리즘으로 되어 있고,  
 • Merge Sort는 재귀적인 알고리즘과 반복적인 알고리즘이 있다.

정답 : ④



