

2번.

우선 순위 큐

데이터들이 우선 순위를 가지고 있고 들어간 순서에 상관없이 우선 순위가 높은 데이터가 먼저 나가는 자료구조

힙으로 가장 효율적인 구조의 우선 순위 큐를 구현할 수 있다.

힙정렬

최대힙이나 최소힙을 구성해 하나 씩 요소를 꺼내서 배열의 뒤에서 부터 저장한다.

힙은 트리 구조이므로 하나의 요소를 삽입하거나 삭제할 때  $O(\log n)$  의 시간 복잡도를 갖는다.

힙 정렬은  $n$ 개의 요소를 힙에 삽입하여 힙을 구성하고  $n$ 개의 요소를 힙에서 삭제하므로

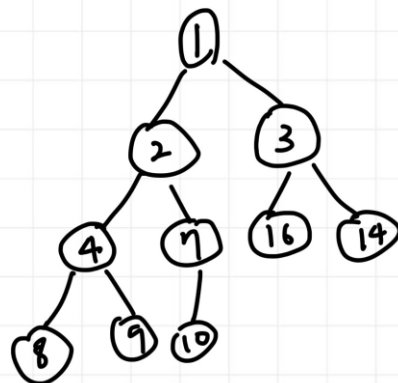
$n \log n + n \log n = 2n \log n$ , 따라서  $O(n \log n)$  의 시간 복잡도를 갖는다.

힙정렬의 장점

최악의 경우에도  $O(n \log n)$  의 시간 복잡도를 보장한다.

추가 메모리를 사용하지 않는다.

1주차 2시간 3번



4번.

카운팅 정렬

효율 : 적은 개수의 숫자를 정렬할 때 가장 빠르며, 비교를 하지 않는 정렬이다.  $n$ 개의 원소가 있고, 최대값이  $k$ 라면 시간 복잡도는  $O(n+k)$ 이다.

제한 : 최대값인  $k$ 가 엄청 크다면, 많은 메모리 공간이 필요하다.

기수 정렬

효율 : 입력 리스트가  $n$ 개의 정수를 가지고, 정수가  $d$ 개의 자릿수를 가지고 있다면 시간 복잡도는  $O(dn)$ 이다. 자릿수가 작을수록 효율이 좋다.

제한 : 키 값이 숫자로 표현되어야만 적용 가능하다. 한글, 한자 등은 매우 많은 버킷이 필요하여 적용 불가능하다.