



CS 스터디(정렬)

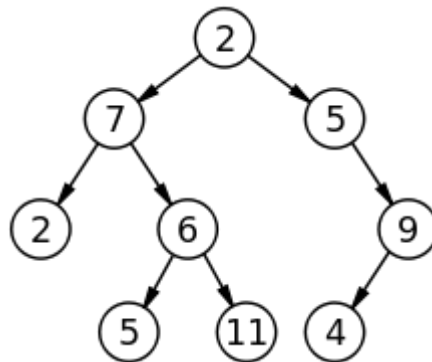
힙 정렬

병합 정렬(Merge Sort)와 퀵 정렬(Quick Sort)만큼 빠르다!

힙이 뭔데?

그 전에 **이진 트리(Binary Tree)**란?

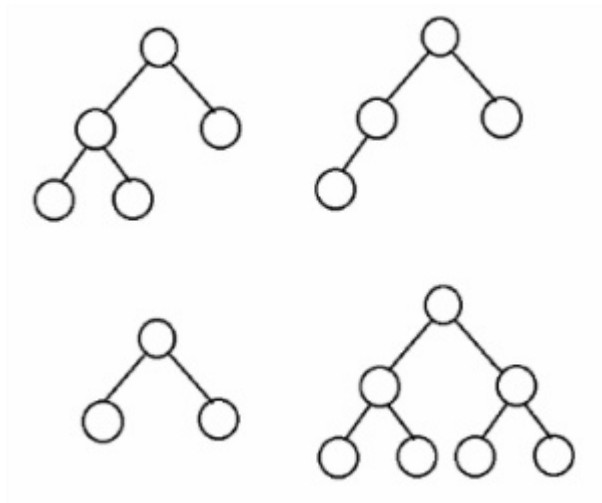
- 각각의 노드가 최대 두 개의 자식 노드를 가지는 트리 자료 구조



https://ko.wikipedia.org/wiki/이진_트리

또! 그 전에 **완전 이진 트리(Complete Binary Tree)**란?

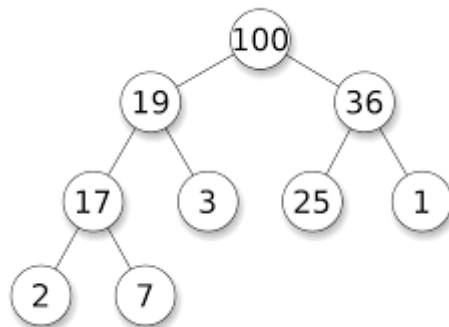
- 마지막 레벨을 제외한 모든 레벨이 완전히 채워져 있고, 마지막 레벨의 모든 노드는 가능한 한 가장 왼쪽에 있는 트리



<https://jomuljomul.tistory.com/entry/완전이진트리Complete-Binary-Tree란>

그래서 **힙**은?

- 최솟값이나 최댓값을 빠르게 찾아내기 위해 완전 이진트리를 기반으로 하는 트리



https://en.wikipedia.org/wiki/Binary_heap

힙 정렬은 어떻게?

- 힙에 데이터 삽입
- 힙에서 데이터 삭제

힙 정렬의 성능

힙의 데이터 저장 및 삭제 $O(\log n)$

→ 정렬과정에서 대상의 수가 n 개라면? $O(n \log n)$

계수 정렬

빠르다! $O(n)$

단순하게 크기를 세는 알고리즘 → 위치를 바꿀 필요가 없다

근데 왜 counting sort 안 쓰고 quick sort!?

→ 대부분의 상황에서 엄청난 메모리 낭비

ex) $[0, 2, 0, 100, 2, 0]$ 을 정렬하기 위해 3~99까지 무의미한 순회를 함

따라서 정렬하는 숫자가 **특정한 범위** 안에 있을 때 사용!

최댓값을 k 라 할 때 $O(n + k)$

기수 정렬

계수 정렬과 마찬가지로 비교연산을 하지 않음!

가장 큰 숫자의 자리수가 d 라면 $O(dn)$

계수 정렬에서 최댓값을 k 라 하면 기수 정렬은 $k = 9$ 이고 자리수별로 counting sort를 실행