자료 구조 스터디 06

2021-2 KCA

우선순위 큐 / Heap

본 ppt의 자료는 Pt.J님의 자료와 김성열 교수님의 강의 및 여러 블로그를 참고하였음을 밝힙니다.

우선순위 큐 (priority queue)

- 일반적인 큐는 선입선출.
- 우선순위 큐는 각 원소들이 우선순위(priority)를 가진다.
  즉, 원소를 push할 때 우선순위를 지정해 주고, 원소를 pop할 때 우선순위대로 나온다.

- 우선순위 큐는 힙, 배열, 연결리스트로 구현 가능하지만, 주로 힙으로 구현(시간복잡도 이점)
  힙 구현 시 삽입, 삭제 O(logN)

- 필요한 최소한의 함수
  <큐가 비었는지 확인> <우선순위를 지정하여 큐에 추가> <가장 높은 우선순위 원소를 제거하고 반환>

우선순위 큐의 구현

- 1. 배열
  우선순위가 높은 순서대로 배열의 가장 앞 부분에 넣는다.
  원소를 넣을 때 우선순위를 비교해서 정렬된 위치에 끼워 넣는다.
  반환은 맨 앞 인덱스
  삽입 : O(n) 삭제 : O(1)

- 2. 연결리스트
  우선순위가 높은 순으로 연결한다.
  삽입 : O(n) 삭제 : O(1)

- 3. 힙
  시간복잡도가 이진 트리의 높이만큼 걸림.
  O(logN)

우선순위 큐를 구현하는 표현 방법	삽입	삭제
순서 없는 배열	O(1)	O(n)
순서 없는 연결 리스트	O(1)	O(n)
정렬된 배열	O(n)	O(1)
정렬된 연결 리스트	O(n)	O(1)
힙(heap)	O(logn)	O(logn)

https://gmlwjd9405.github.io/2018/05/10/data-structure-heap.html

이진 트리

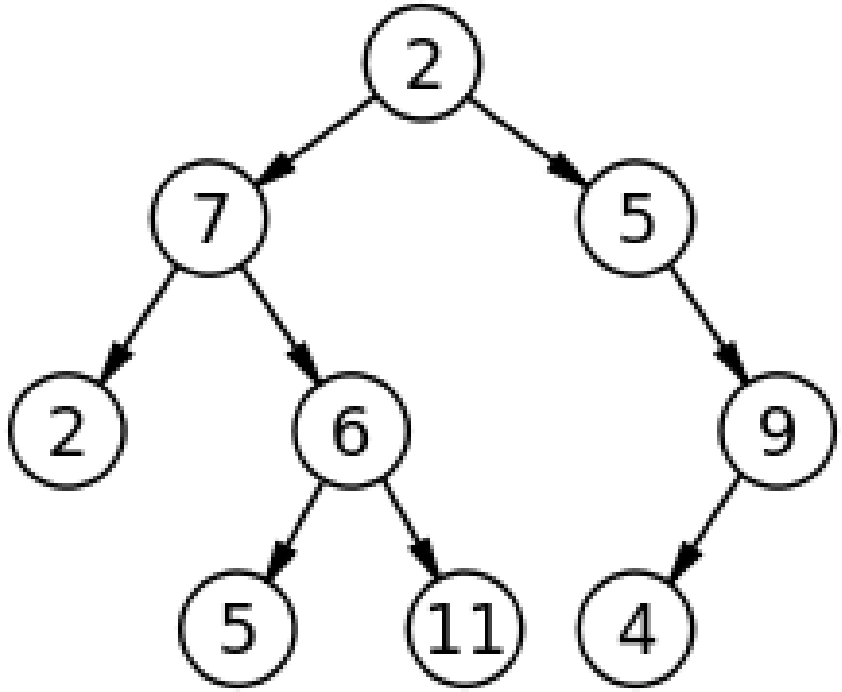
⇒ 각각의 노드가 최대 두 개의 자식 노드만 가지는 트리

- 완전 이진 트리

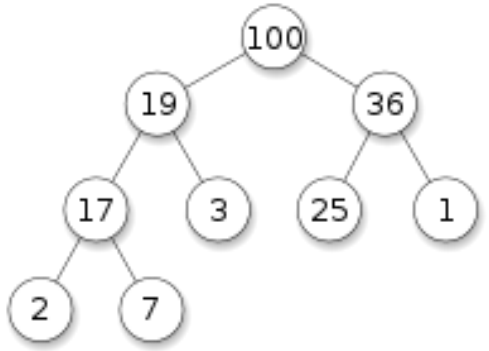
⇒ 마지막 레벨 이외에는 모두 노드가 채워져 있으며, leaf 노드가 왼쪽 - 오른쪽 순으로 채워지는 트리

- 균형 이진 트리

⇒ 양 쪽의 트리 높이 차가 1 이하



힙 (Heap)



Max Heap

- 완전 이진 트리

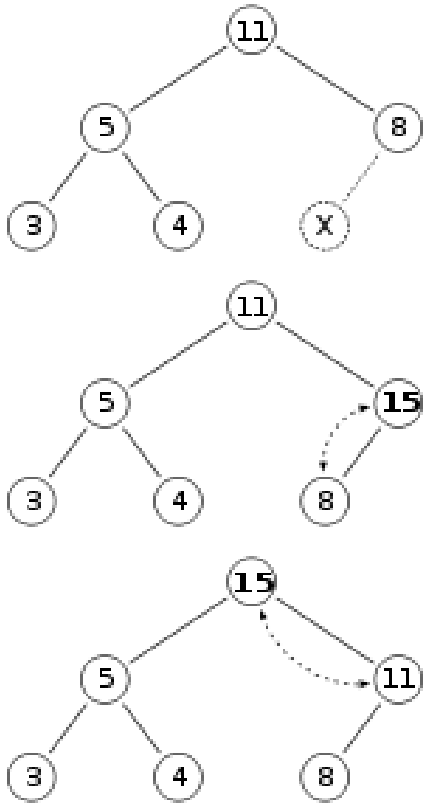
<힙 조건> (노드에 표기되는 값을 우선순위로 본다.)
1) root 노드는 항상 우선순위가 가장 높다.
2) 각 노드의 우선순위는 자식 노드의 우선순위보다 크거나 같다.

- 첫 번째 우선순위를 찾기 매우 편리. (중간을 찾기는 힘들지만..)

- Min Heap (최소 힙) : 루트 노드로 올라갈수록 값이 작음. 값이 작을수록 우선순위가 높음.
- Max Heap (최대 힙) : 루트 노드로 올라갈수록 값이 큼. 값이 클수록 우선순위가 높음.

⇒ 최소 힙, 최대 힙 모두 <힙 조건>을 만족!!

Heap Insert(Push), Extract(Pop)

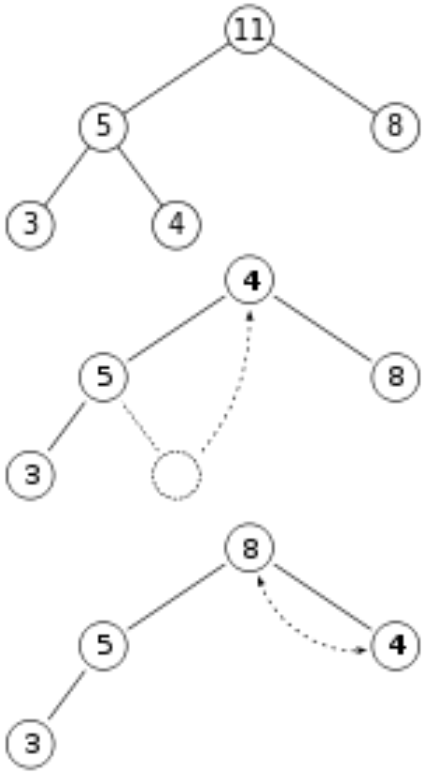


예) 15 Insert

1. 맨 마지막에 삽입한다.
2. 부모 노드와 비교해서 힙 조건을 만족할 때까지 위치를 바꾼다.

시간복잡도 : O(logN)

임의의 값 삽입에 대한
평균 시간복잡도 : O(1)



예) 11 Pop

1. 맨 마지막 노드를 빈 위치로 옮긴다.
2. 두 자식 노드 중 우선순위가 높은 것과 비교한다.
3. 힙 조건을 만족할 때까지 자식 노드와 위치를 바꾼다.

시간복잡도 : O(logN)

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

Heap Search, Delete

- Search

시간복잡도 O(n)

- Delete

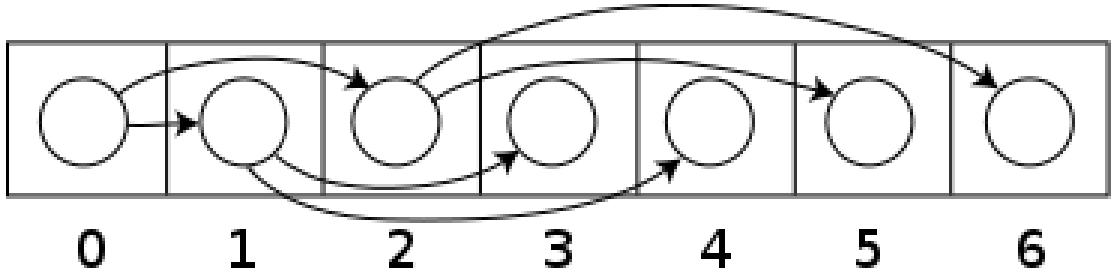
특정 노드 Search 후 삭제,

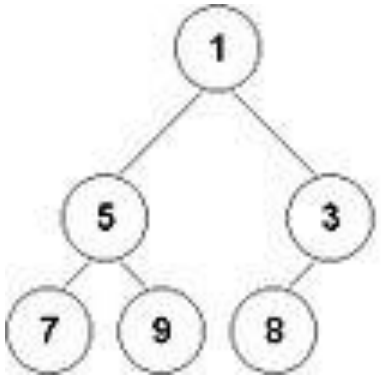
마지막 노드로 교체

힙 조건을 만족할 때까지 부모 노드 or 자식 노드와 위치 바꾸기.

힙 구현

- 힙은 보통 배열로 구현
- 완전 이진 트리 이므로 배열의 칸에 1대1 대응 가능.
- 자식 노드의 인덱스 :
  (부모 노드 + 1) * 2
  (부모 노드 + 1) * 2 - 1





Node	1	5	3	7	9	8
Index	0	1	2	3	4	5

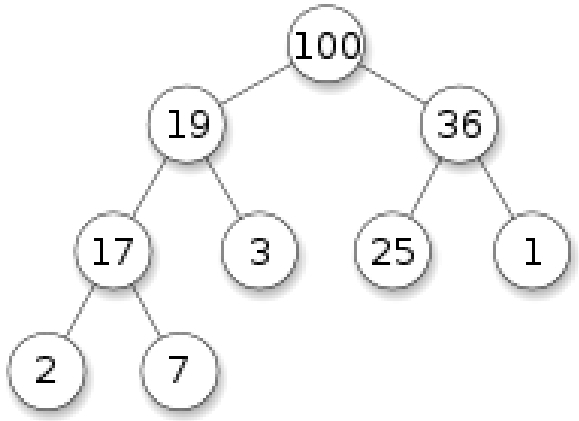
[참고] 힙 정렬
힙에서 우선순위대로 하나씩 뽑으면 된다.
시간복잡도 : O(NlogN)

Heap vs. BST

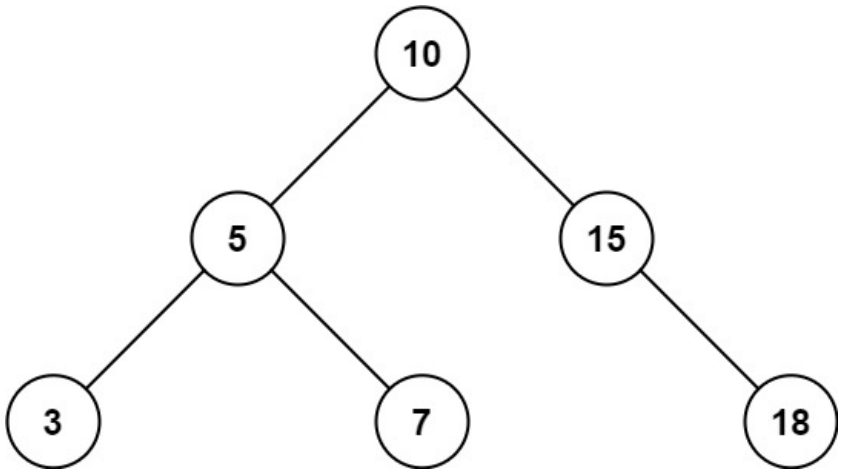
- 힙은 상하관계로 정의되고, BST는 좌우관계로 정의된다.

- 힙은 root 노드가 가장 우선순위가 높고,
  모든 노드는 자식 노드보다 우선순위가 높거나 같아야 한다. (좌우 우선순위는 상관없음, 중복 가능)
- 장점 : 평균 임의 노드 삽입이 O(1), <우선순위를 뽑을 때 주로 사용>

- BST는 왼쪽 노드의 값이 오른쪽보다 항상 작아야 한다. (상하 크기는 상관없음, 중복 불가)
- 장점 : 모든 노드에 대한 검색이 O(logN) 보장. <정렬을 유지하는 데이터를 원할 때 주로 사용>



Max Heap



BST