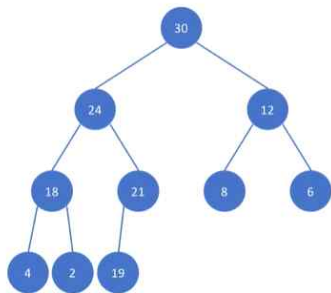


22강, 힙(Heaps)

1) 힙(Heap)이란?

이진 트리의 한 종류 (이진 힙-binary heap)

1. 루트(root) 노드가 언제나 최댓값 또는 최솟값을 가짐
: 최대 힙(max heap), 최소 힙(min heap)
2. 완전 이진 트리여야 함



최대 힙의 예

2) 이진 탐색 트리와의 비교

1. 원소들은 완전히 크기 순으로 정렬되어 있는가? (이진 탐색 트리만 가능)
2. 특정 키 값을 가지는 원소를 빠르게 검색할 수 있는가? (이진 탐색 트리만 가능)
3. 부가의 제약 조건은 어떤 것인가? (힙은 완전 이진 탐색 트리여야함)

3) 최대 힙(Max Heap)의 추상적 자료구조

연산의 정의

`__init__()` : 빈 최대 힙을 생성

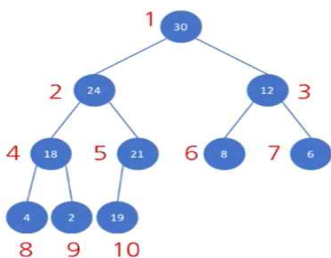
`insert(item)` : 새로운 원소를 삽입

`remove()` : 최대 원소(root node)를 반환 (그리고 동시에 이 노드를 삭제)

4) 데이터 표현의 설계

: 배열을 이용한 이진 트리의 표현

배열을 이용한 이진 트리의 표현



노드 번호 m 을 기준으로

- 왼쪽 자식의 번호: $2 * m$
- 오른쪽 자식의 번호: $2 * m + 1$
- 부모 노드의 번호: $m // 2$

완전 이진 트리이므로

- 노드의 추가/삭제는 마지막 노드에서만

0	1	2	3	4	5	6	7	8	9	10
-	30	24	12	18	21	8	6	4	2	19

5) 최대 힙에 원소 삽입

1. 트리의 마지막 자리에 새로운 원소를 임시로 저장
2. 부모 노드와 키 값을 비교하여 위로, 위로, 이동

6) 최대 힙에 원소 삽입- 복잡도

원소의 개수가 n 인 최대 힙에 새로운 원소 삽입
-> 부모 노드와의 대소 비교 최대 회수: $\log(2)n$
최악 복잡도 $O(\log n)$ 의 삽입 연산

7) 문제

초기 코드에 주어진 class MaxHeap 에 최대 힙에 새로운 원소를 추가하는 연산인 insert() 메서드의 구현을 완성하세요.

[참고 1] solution() 함수의 구현은 그대로 두세요. 이것을 없애면 테스트가 되지 않습니다.

[참고 2] 코드 실행 을 눌렀을 때 통과하는 것은 아무런 의미가 없습니다.

```
class MaxHeap:

    def __init__(self):
        self.data = [None]

    def insert(self, item):
        self.data.append(item)
        i=len(self.data)-1
        while i>1:
            if self.data[i]>self.data[i//2]:
                self.data[i],self.data[i//2]=self.data[i//2],self.data[i]
                i=i//2
            else:
                break

    def solution(x):
        return 0
```