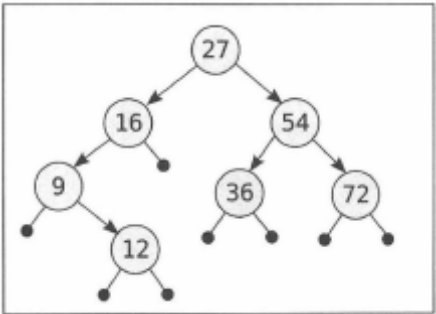


# 트리

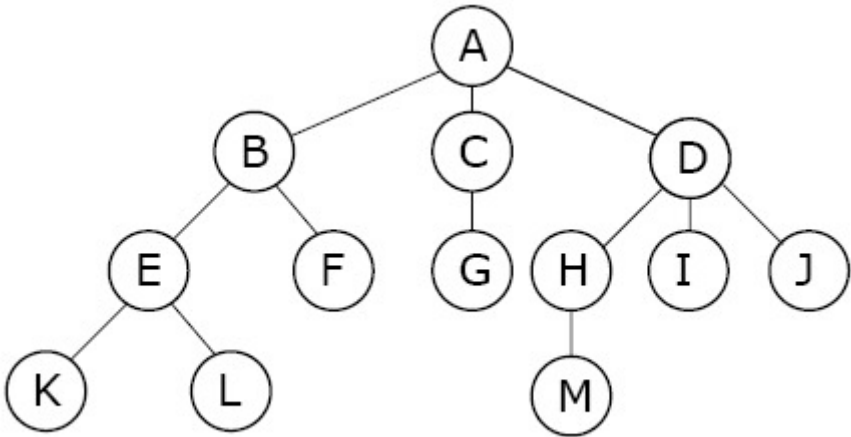
통계 수정 삭제

deepredk · [방금 전](#) · [비공개](#)

🍷 0



# 트리



# 소개

- 계층 구조를 나타내는 자료구조다.

- 탐색형 자료구조로도 유용하게 쓰인다.
  - \* 예를 들어 이진 검색 트리를 사용하면  $O(\log N)$ 의 시간복잡도로 데이터를 찾아낼 수 있다

## 용어

- Node: 자료가 저장되는 단위
- 간선: 노드와 노드에 간선을 그림으로써 노드의 계층 구조를 나타냄
- 부모: 상위 노드
- 자식: 하위 노드
- 형제: 부모가 같은 노드
- Root: 최상위 노드
- Leaf: 자식이 하나도 없는 노드
- 깊이: 어떤 노드에 도달하기 위해 거쳐야 하는 간선의 수
- 높이: 가장 깊숙히 있는 노드의 깊이

## 구현

- 클래스, 구조체 등으로 Node를 구현한다.
- 자바 기준으로 `class Node { Data data; Node parent, List children; }`
  - \* 데이터를 데이터, 부모 노드 포인터, 자식 노드들 포인터

## 순회

- 루트의 자식 노드들은 또 하나의 서브트리다. 이 재귀적 특성을 이용하여 순회한다.
- 자바 기준으로 아래와 같이 구현할 수 있다.

```
void printNodes(Node root) {  
    System.out.print(root.data);  
    for (Node child : root.children) {  
        printNodes(child);  
    }  
}
```

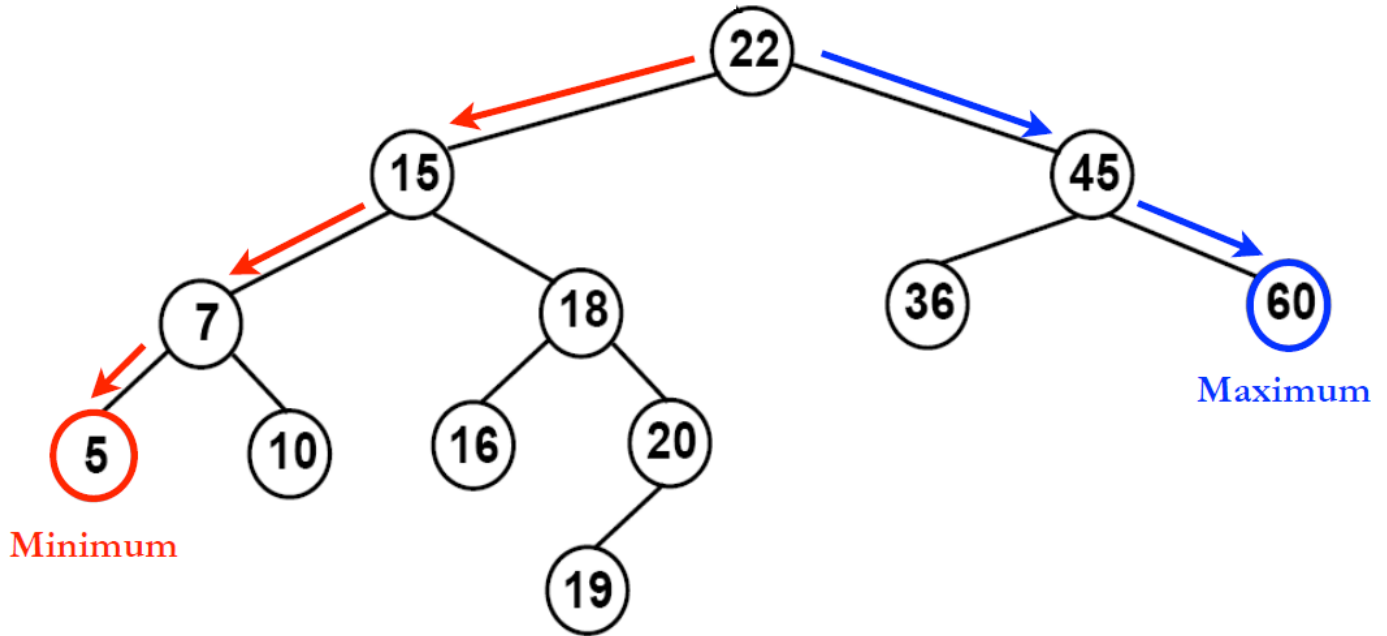
## 이진 트리

### 소개

- 각 노드가 왼쪽, 오른쪽 자식 노드만 가질 수 있는 트리
- 순회 방법으로 전위, 중위, 후위 순회가 있다.

- 전위 순회: 루트 -> 왼쪽 서브트리 -> 오른쪽 서브트리 순으로 방문
- 중위 순회: 왼쪽 서브트리 -> 루트 -> 오른쪽 서브트리 순으로 방문
- 후위 순회: 왼쪽 서브트리 -> 오른쪽 서브트리 -> 루트 순으로 방문

## 이진 검색 트리



## 소개

- '이진 탐색' 알고리즘의 아이디어를 차용하여 사용하는 트리
- 왼쪽 서브트리에는 더 작은 값을 가진 노드들이,
- 오른쪽 서브트리에는 더 큰 값을 가진 노드들이 들어간다

## 다양한 연산

### 순회

- 중위 순회하면 오름차순 정렬된 목록을 얻을 수 있다.

### 검색

- 찾고싶은 값이 73이고, 루트 노드가 100이라면?  
\* 왼쪽 서브트리를 방문해야할 것이다.
- 이런 식으로 재귀적으로 탐색해나가면 해당 값을  $O(\log N)$ 의 시간복잡도로 찾아낼 수 있다.

## 추가

- '검색' 연산으로 새 원소가 들어갈 위치를 쉽게 알 수 있다.
  - 왼쪽 서브트리, 오른쪽 서브트리, 왼쪽 서브트리를 방문하다가
  - 어라, 왼쪽 서브트리를 방문해야 하는데 왼쪽 서브트리가 비어있네?
  - 그럼 그 비어있는 쪽에 추가하면 된다.

## 삭제

- 자식이 없는 노드라면 그냥 삭제한다.
- 자식을 하나만 가진 노드라면 해당 노드의 부모 노드와 자식 노드를 서로 연결하고 자신은 삭제한다.
- 자식을 둘 모두 가진 노드라면
  - 두 서브트리를 합친 새로운 이진 검색 트리를 만들고
  - 지울 노드의 자리에 새로운 이진 검색 트리의 루트를 끼워넣는다.



김진홍

## 0개의 댓글

댓글을 작성하세요

댓글 작성