

## B+ Tree

### [입력]

삽입 – key, data

삭제 – key

### [출력]

내부 노드 – key만 출력

리프 노드 – key와 data 모두 출력

출력 – Traverse(node\* root)

### [기준]

1. 노드에 존재할 수 있는 키의 최대 개수는 3개이다.
2. 부모의 키 값과 같은 값은 왼쪽 서브 트리에 위치한다.
3. 트리의 리프 노드에 존재하는 키들은 오름차순 정렬한다.
4. 삭제하고자 하는 키는 무조건 트리 내에 존재한다. 삽입하고자 하는 키는 무조건 트리 내에 존재하지 않는다.
5. Top-Down 방식(Introduction to Algorithms 기준)

타겟 노드 = K가 존재할 가능성이 있는 노드

6. 삽입 – 타겟 노드가 꽉 차 있으면(키의 개수 = 3) 분리한다.
7. 삭제 – 타겟 노드의 key의 개수가 1개면 2개 이상으로 만들고 방문한다.
  - A. K를 삭제하기 위한 삭제 과정 중 리프 노드가 아닌 내부 노드에 K와 같은 값의 키가 있어도 삭제하지 않는다.
  - B. 타겟 노드가 리프 노드가 아닐 때

형제 노드에서 값을 빌려올 때는 무조건 오른쪽 형제 노드에서 가져오고, 오른쪽 형제 노드에서 가져올 수 없다면(타겟 노드와 오른쪽 형제 노드의 키 개수가 모두 1개일 때), 오른쪽 노드와 타겟 노드를 합친다.

타겟 노드가 형제 노드들 중 제일 오른쪽 노드인 경우 왼쪽 형제에게 빌릴 수 있으면 빌리고, 불가능하면 왼쪽 형제 노드와 합친다.