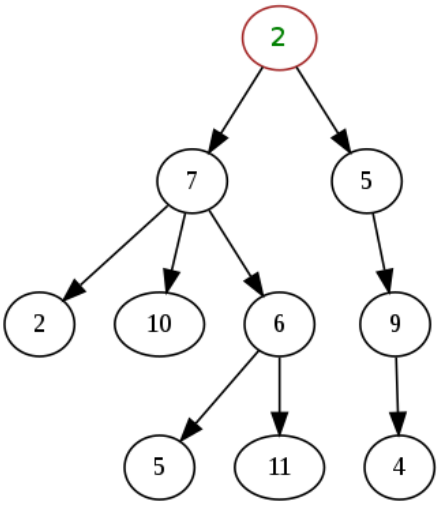
ILE3-016 Tree

연결 리스트의 변형으로 부모 - 자식 구조를 지닌 그래프 형태의 자료구조

각 항목은 Node라고 명칭

순환구조를 지닐 수 없음



Node의 명칭

Root Node

부모를 지니지 않는 가장 높은 위치(최상위)에 존재하는 노드

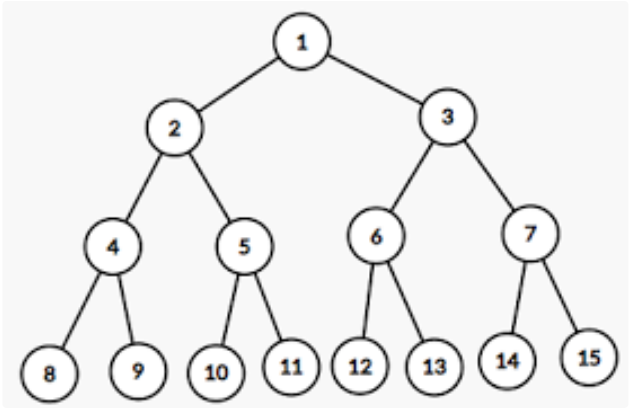
Leaf Node

자식을 지니지 않는 가장 낮은 위치(최하위)에 존재하는 노드

Tree의 종류

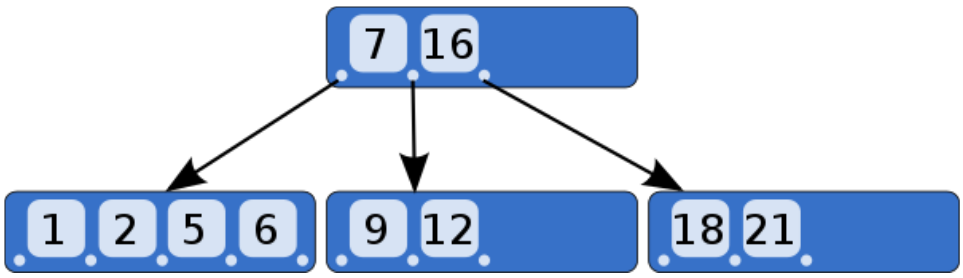
이진 트리

자식 노드를 2개만 지니는 Tree



B-Tree

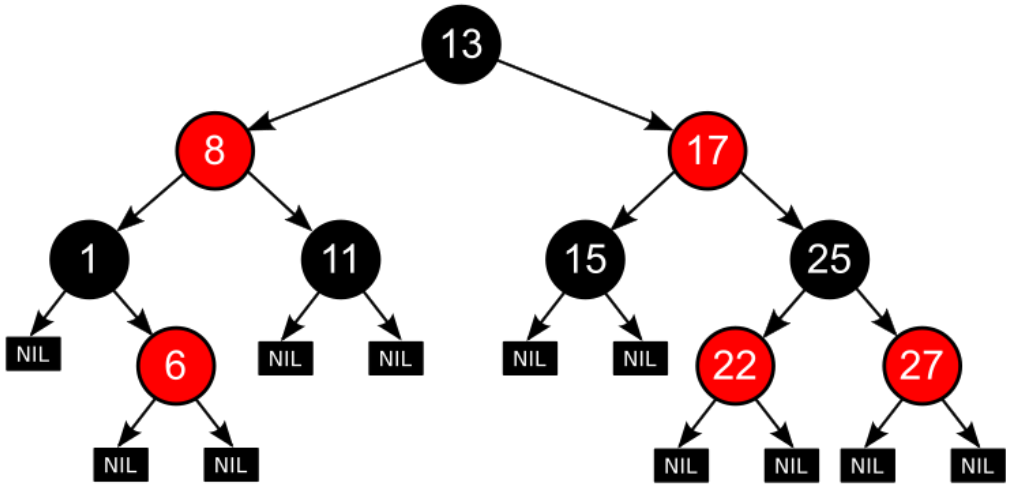
이진 트리를 확장한 구조로 데이터
하나의 노드가 가질 수 있는 자식 노드의 최대 숫자가 2보다 큰 트리
데이터를 정렬 된 상태로 보관하고, Get, Insert, Delete 작업이 모두 `O(logn)`에 처리 가능
일반적으로 Database의 Index로 사용



Red-Black Tree

자가 균형 이진 트리로 불리며, 최악의 경우에도 우수한 성능을 보이는 Tree 구조

- Node는 Black - Red 중 하나의 색상을 지님
- Root Node는 항상 Black
- Leaf Node는 항상 Black
- Red Node의 자식 노드는 항상 Black이며, Red Node는 연달아 표현 될 수 없
- 좌측의 자식 노드는 부모 노드보다 작아야 하며, 우측의 자식 노드는 부모 노드보다 값이 커야 함
- 신규 추가되는 노드는 항상 Red Node

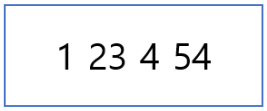


과제

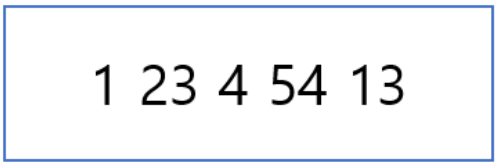
Node에 값을 4개까지 지닐 수 있고 자식 노드는 2개를 지니는 B-Tree를 만들어 봅시다.
중복된 값을 입력하는 경우는 가정하지 않습니다.

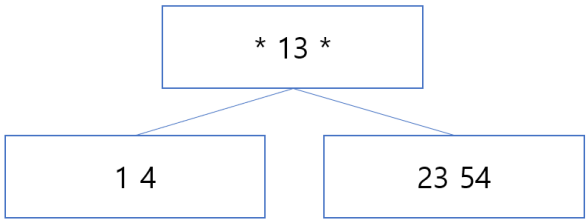
Insert

1. Root Node부터 시작합니다.
2. 현재 노드가 자식 노드를 가지고 있지 않은 경우 현재 노드에 값을 추가합니다.
   a. 값이 추가 되었을 때 Node에 값이 4개 이하인 경우
      해당 값을 저장합니다.



   b. 값이 추가 되었을 때 Node에 값이 5개가 된 경우
      Node 내의 값을 정렬하여 중심 값을 기준값으로 설정하고 자식 노드를 생성하며 좌측에는 기준값보다 작은 값, 우측에는 기준값보다 큰 값을 입력합니다.





3. 현재 노드가 자식 노드를 지니고 있는 경우, 현재 노드의 기준값과 입력값을 비교하여 기준값보다 작은 경우 좌측 노드를, 큰 경우 우측 노드를 현재 노드로 선택하고 2.로 돌아갑니다.

Get

1. Root Node부터 시작합니다
2. 현재 노드가 자식 노드를 지니고 있지 않은 경우
   a. 현재 노드에 입력 값을 지니고 있는지 여부를 리턴합니다.
3. 현재 노드가 자식 노드를 지니고 있는 경우
   a. 현재 노드의 기준값이 입력 값과 동일 한 경우에는 값이 존재한다고 리턴
   b. 현재 노드의 기준값과 입력 값이 다른 경우
      i. 입력 값이 기준 값보다 작은 경우 좌측 노드를 현재 노드로 선택하고 2.로 돌아갑니다.
      ii. 입력 값이 기준 값보다 큰 경우 우측 노드를 현재 노드로 선택하고 2.로 돌아갑니다.

테스트 코드

1. 0 ~ 1,000,000 사이의 임의의 값 100,000개를 중복 없이 추출합니다.
2. 0 ~ 1,000,000 사이의 임의의 값 10,000개를 중복 없이 추출합니다.
3. List에 1.에서 추출한 값을 삽입합니다.
4. B-Tree에 1.에서 추출한 값을 삽입합니다.
5. List의 `in` 연산자를 이용하여 2.에서 추출한 값이 3.에서 만든 List에 존재하는지 판단하고, 해당 과정이 소요 된 시간을 측정합니다.

6. B-Tree에서 2.에서 추출한 값이 3.에서 만든 List에 존재하는지 판단하고, 해당 과정이 소요 된 시간을 측정합니다.

Example

List 내에 특정 값이 존재하는 지 확인하는 방법

```
l = [1, 3, 5, 7, 9]
if 3 in l:
  print('3이 존재합니다.')
```