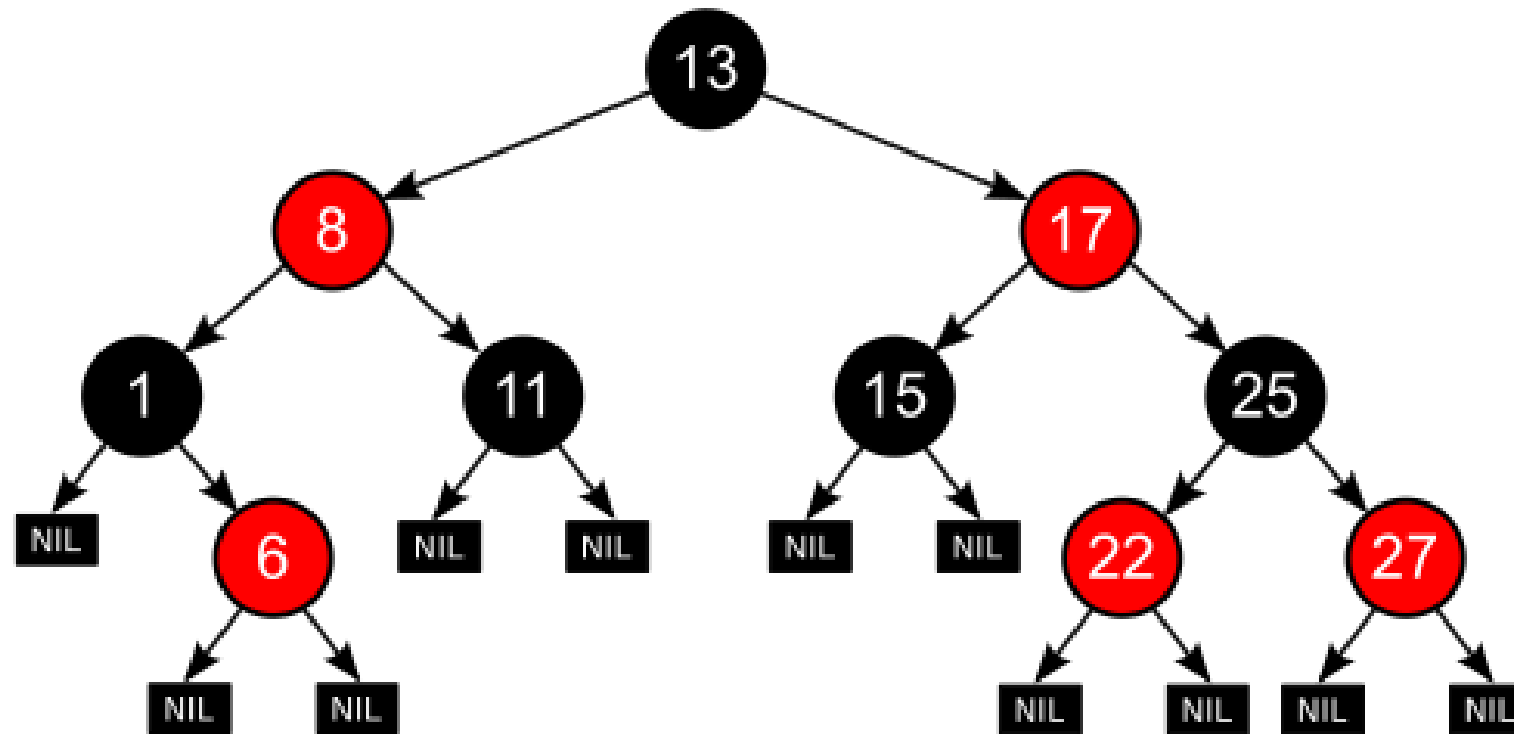


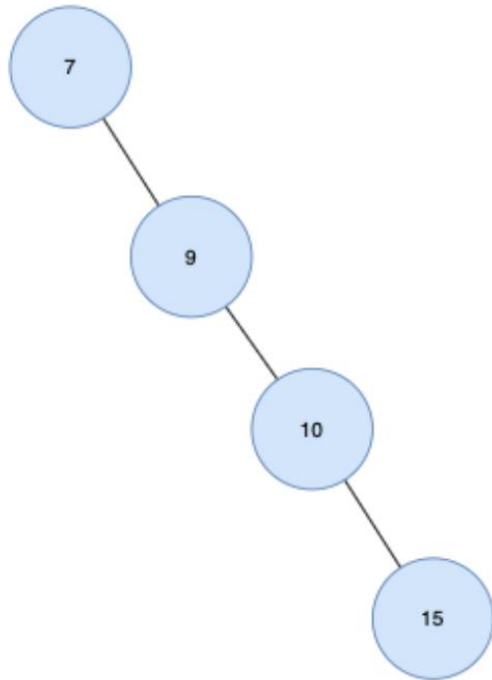
## 2주차 주제 자료구조

RED-BLACK TREE가 무엇인가

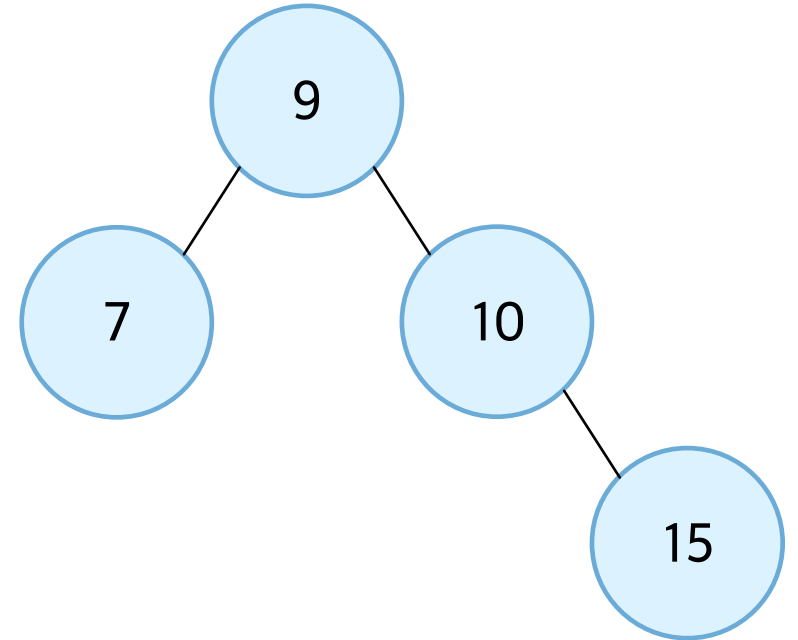


# Red-black tree를 사용해야 하는 이유?

7, 9, 10, 15 저장할 때 편향이진트리가 됨



자가균형이진트리 (AVL, Red-black tree)



문제점: 공간을 많이 소비하고 탐색할 때 시간도 많이 소모된다.

---

# AVL tree보다 Red-black tree를 많이 쓰는 이유

## 참고

<https://ebongzzang.github.io/algorithm/Red-Black-tree-%EA%B7%B8%EB%A6%AC%EA%B3%A0-AVL-tree%EC%99%80%EC%9D%98-%EB%B9%84%EA%B5%90/#avl-tree%EC%99%80-%EB%B9%84%EA%B5%90>

# 우리가 알아야 하는 이유

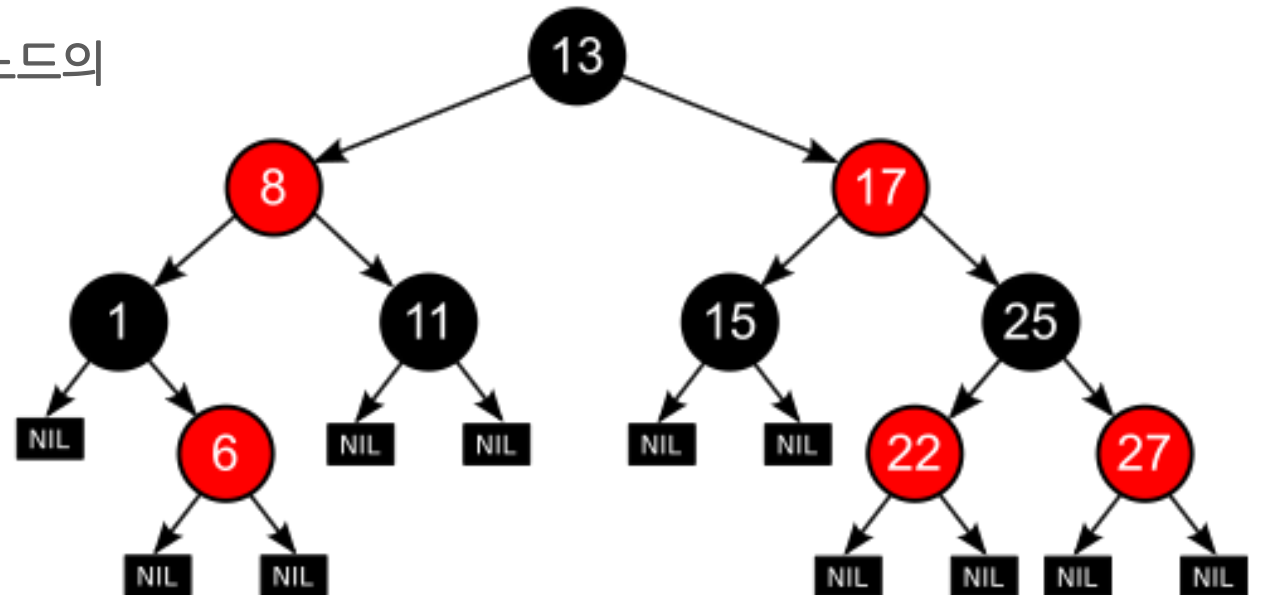
## 자바 <TreeSet, TreeMap>

Red-black tree를 베이스로 작동하며, **검색**과 **정렬**에 적합한 컬렉션 클래스다.

>> 검색과 정렬 외에는 HashMap이 TreeMap보다 더 뛰어나므로  
HashMap을 사용하는 것이 좋다.

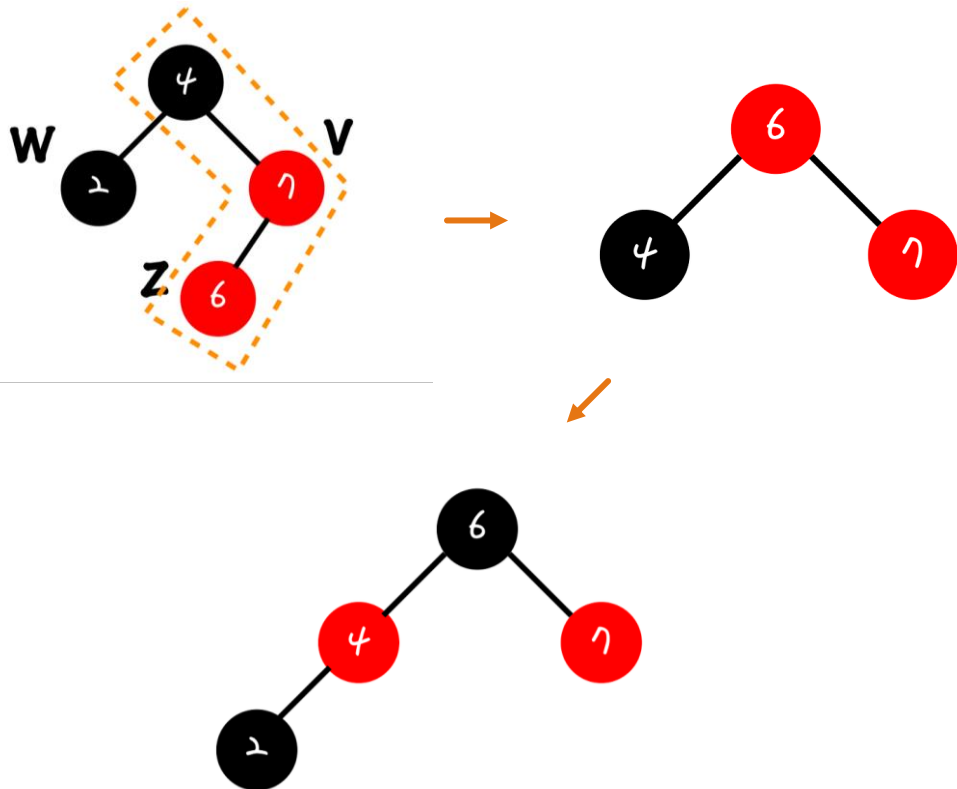
# Red-black tree의 조건

1. 모든 노드는 빨간색 혹은 검은색이다.
2. 루트 노드는 검은색이다.
3. 빨간색 노드의 자식은 반드시 검은색이다. (red - red 가 연속되면 안 된다.)
4. 모든 리프 노드에서 루트 노드까지의 경로 중 검은 노드의 개수가 같다.

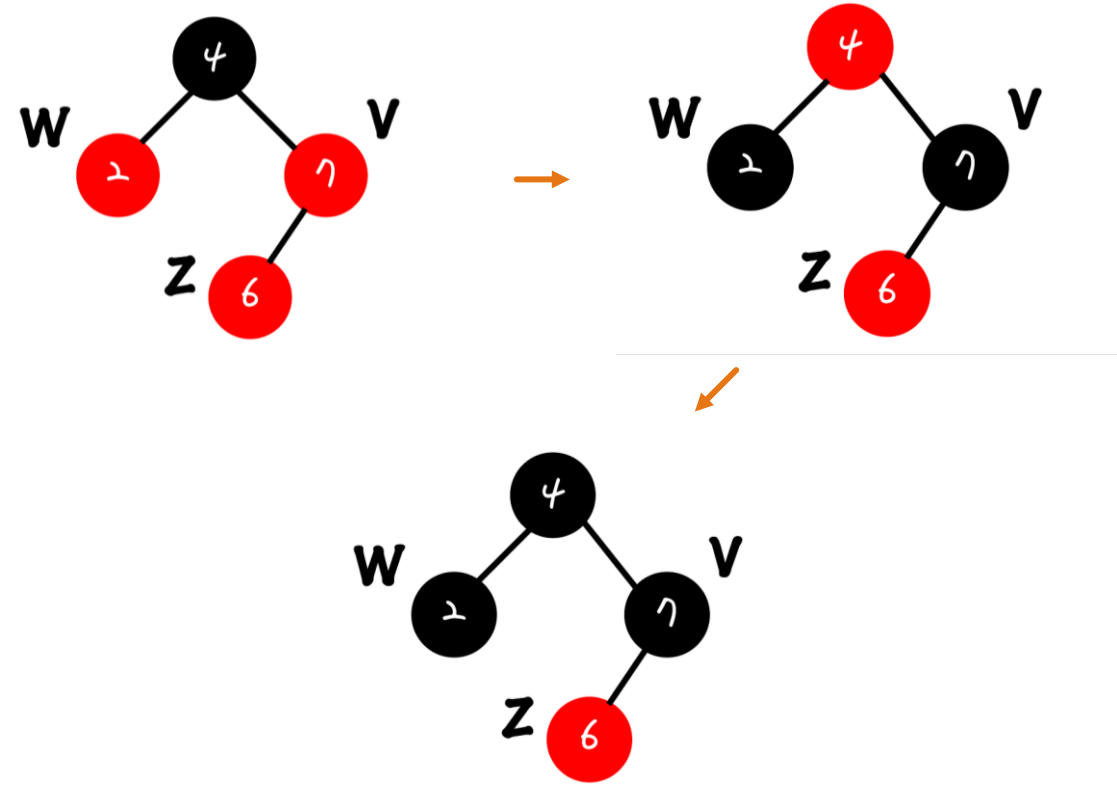


# Red-black tree의 동작

## ■ Restructuring



## ■ Recoloring (삼촌 노드가 레드일 때)



---

# Red-black tree 시뮬레이션

<https://www.cs.usfca.edu/~galles/visualization/RedBlack.html>

# 끝

- 자료구조는 방대하다..

도움 받은 곳

<https://jwdeveloper.tistory.com/280>

<https://jaehoney.tistory.com/283>

