

데이터구조 7장

07-1. 이진탐색트리 개념

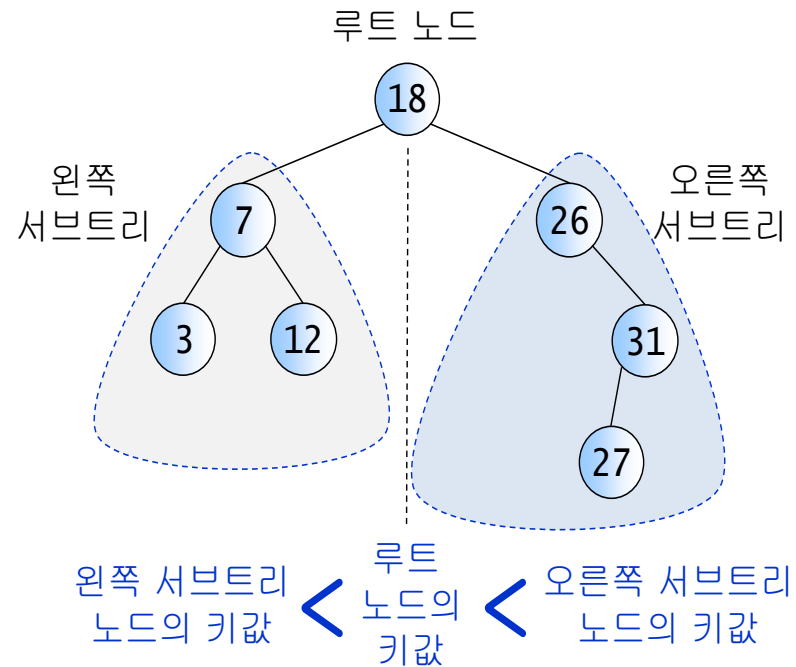
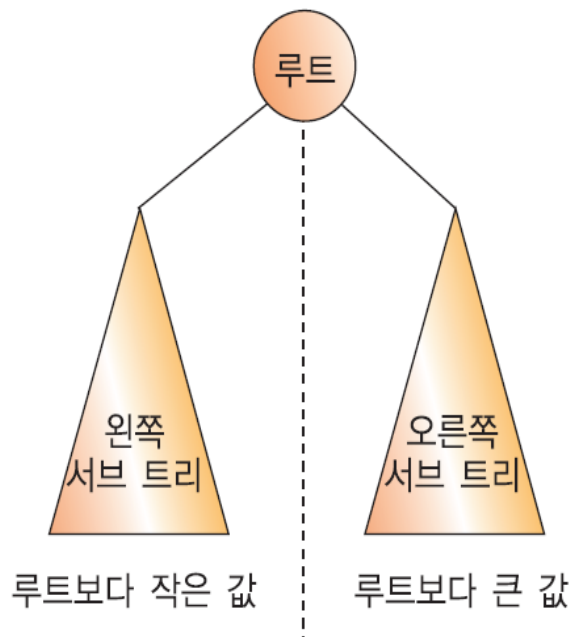
- [illegible]



이진탐색트리의 정의

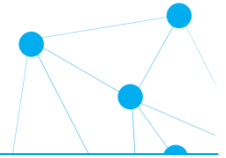


- 탐색작업을 효율적으로 하기 위한 자료구조
 - $\text{key}(\text{왼쪽서브트리}) \leq \text{key}(\text{루트노드}) \leq \text{key}(\text{오른쪽서브트리})$



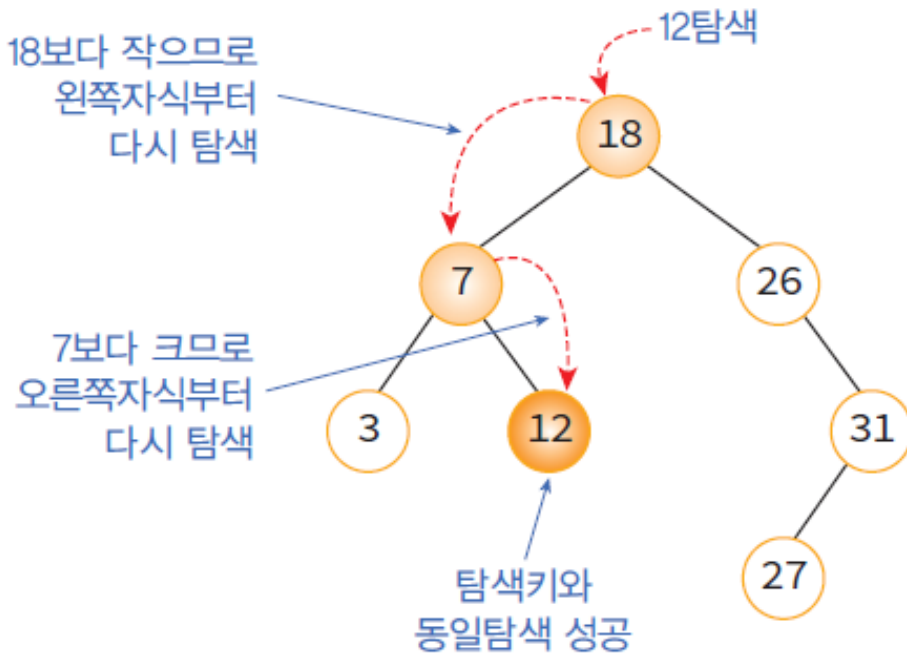
07-2. 이진탐색트리 연산

이진탐색트리의 탐색연산



- 탐색 연산

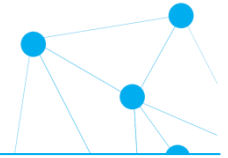
- 비교한 결과가 같으면 탐색이 성공적으로 끝난다.
- 키 값이 루트보다 작으면 ➔ 왼쪽 자식을 기준으로 다시 탐색
- 키 값이 루트보다 크면 ➔ 오른쪽 자식을 기준으로 다시 탐색



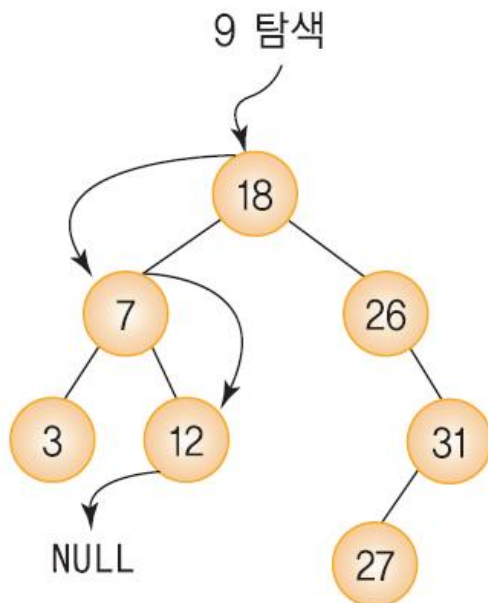
이진탐색트리의 생성,
삽입, 삭제 등 모든 연
산은 탐색에서 출발

결국 루트에서부터 비
교해서 새로운 노드를
하나씩 탐색하는 과
정이 이진탐색트리를
생성하는 과정이 된다.

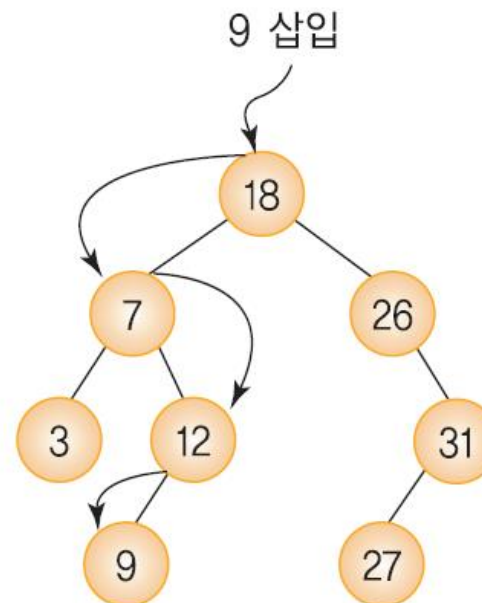
이진탐색트리의 삽입연산



- 먼저 탐색을 수행
 - 탐색에 실패한 위치가 바로 새로운 노드를 삽입하는 위치



(a)



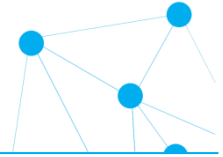
(b)

이진탐색트리의 삭제연산

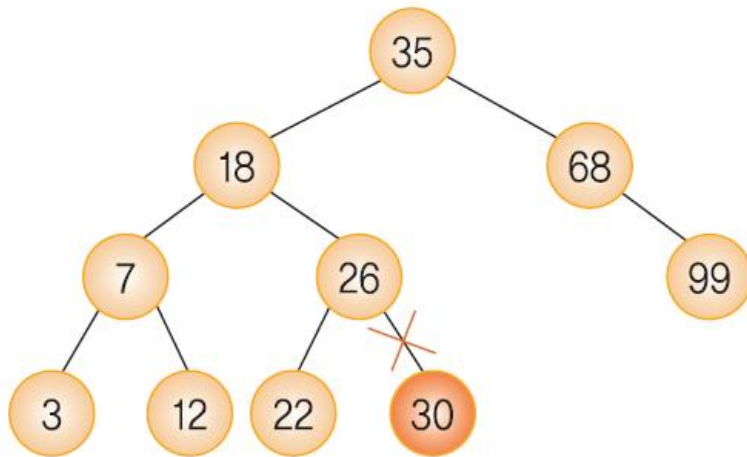


- 노드 삭제의 3가지 경우
 1. 삭제하려는 노드가 단말 노드일 경우
 2. 삭제하려는 노드가 하나의 왼쪽이나 오른쪽 서브 트리중 하나만 가지고 있는 경우
 3. 삭제하려는 노드가 두 개의 서브 트리 모두 가지고 있는 경우

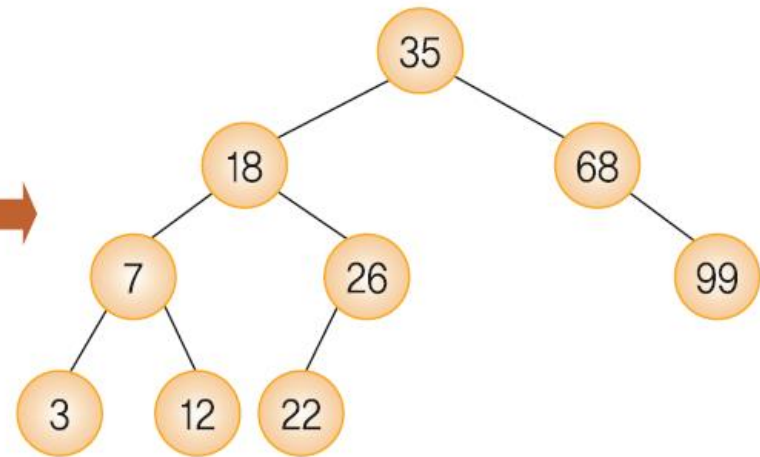
Case 1: 단말 노드 삭제



- 단말노드의 부모노드를 찾아서 연결을 끊으면 된다.



(a) 삭제 전

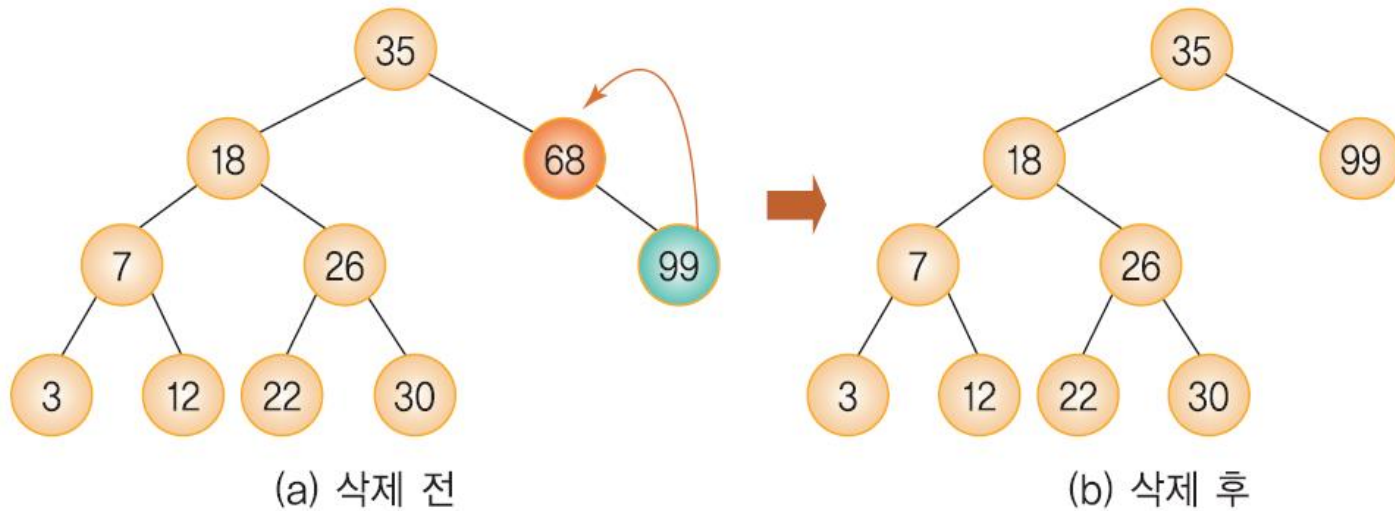


(b) 삭제 후

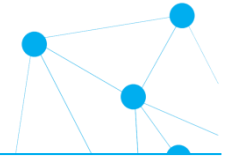
Case 2: 자식이 하나인 노드 삭제



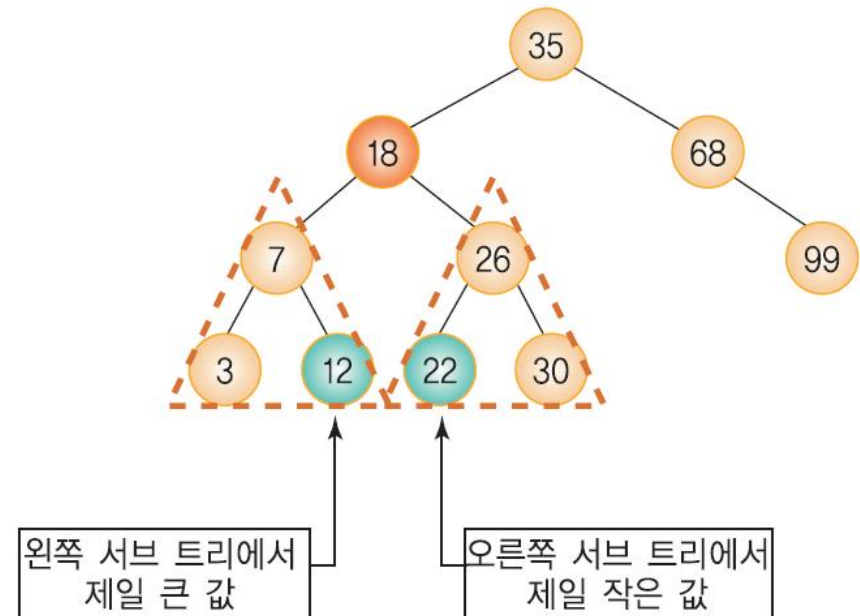
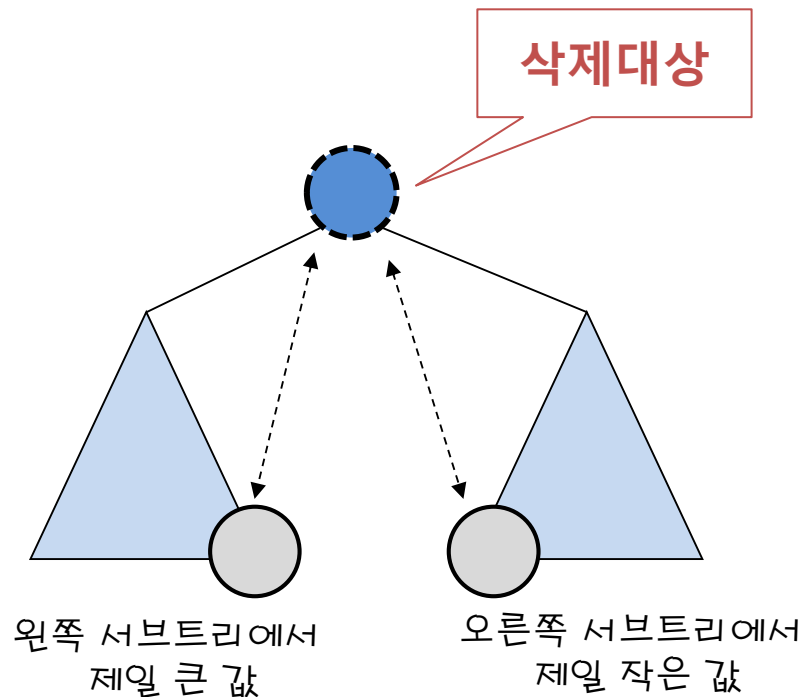
- 노드는 삭제하고 서브 트리는 부모 노드에 붙여줌

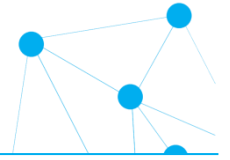


Case 3: 두 개의 자식을 가진 노드 삭제



- 가장 비슷한 값을 가진 노드를 삭제노드 위치로 가져옴
- 후계 노드의 선택





- Case3: 노드 18 삭제 과정

