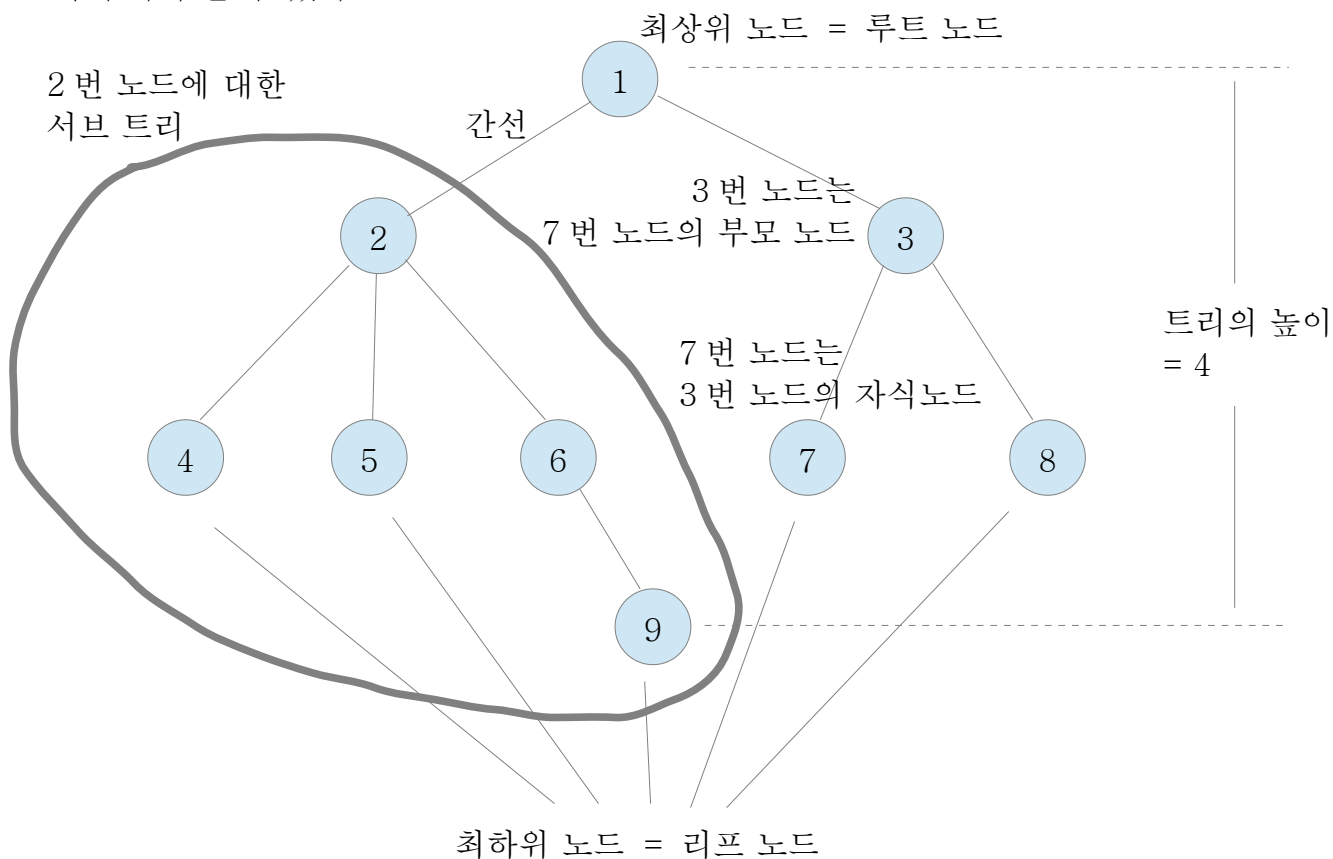
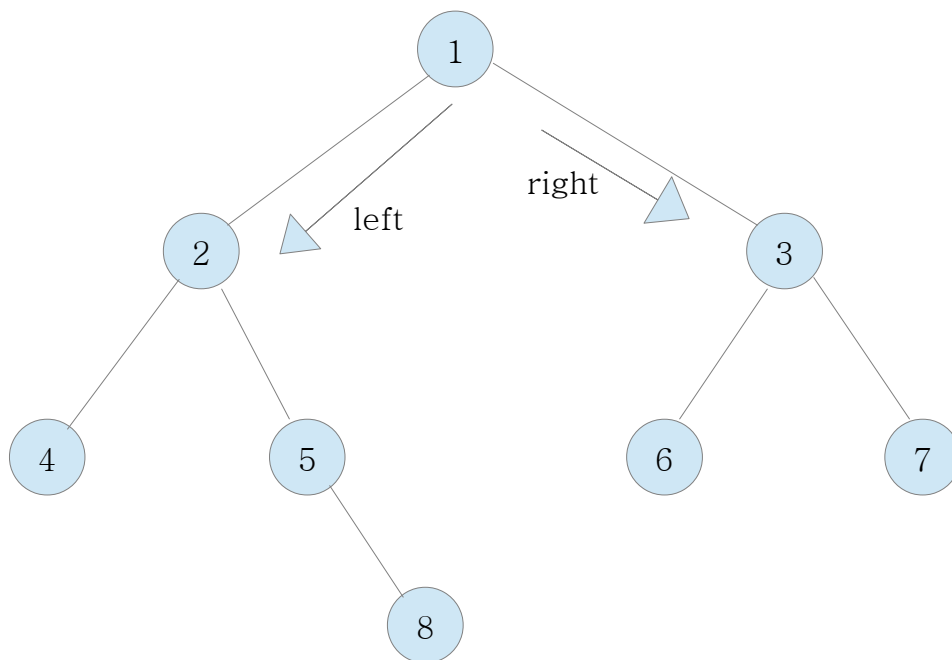


1. 트리: 노드에 위·아래 계층이 주어진 그래프. 최상위 노드는 한 개 존재하고, 각 노드는 부모 노드를 최대 하나 둘 수 있다.



그리기 1: 트리의 구성

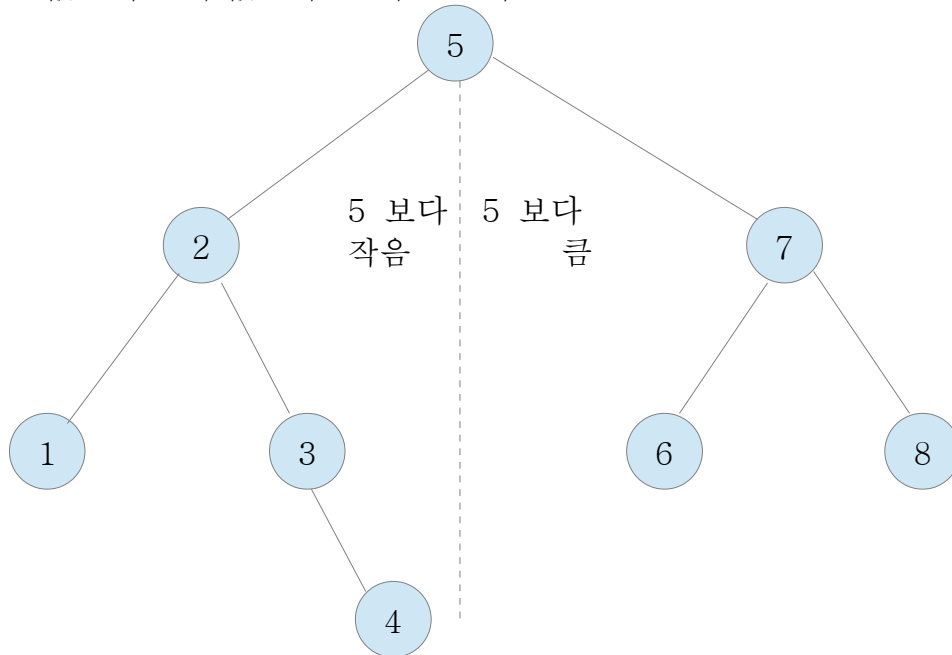
2. 이진트리: 모든 노드에 대해 자식을 최대 둘까지만 가질 수 있는 트리



그리기 2: 이진트리의 예시

한 노드에 대해 자식 노드는 최대 둘까지만 있으므로 각각을 왼쪽 자식 노드, 오른쪽 자식 노드라고 부르도록 한다.

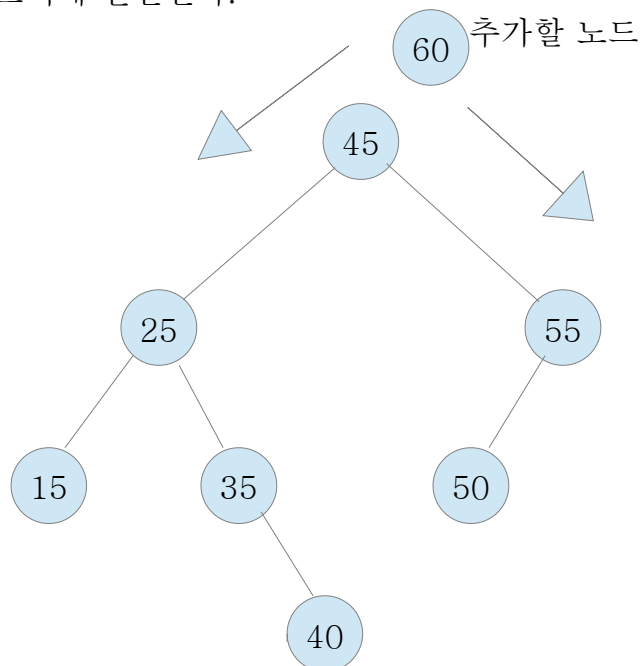
3. 이진검색트리(Binary Search Tree): 왼쪽 서브 트리의 모든 값은 부모의 값보다 작고, 오른쪽 서브 트리의 모든 값은 부모의 값보다 큰 이진 트리



그리기 3: 이진검색트리의 예시

4. 이진검색트리에서 값을 **조회**하기: 루트 노드에서 시작하여 해당 노드의 값과 찾고자 하는 값을 비교하여 왼쪽 자식 노드 혹은 오른쪽 자식 노드로 내려간다. 리프 노드까지 내려갔지만 찾고자 하는 값이 없으면 해당 트리에는 해당 값이 없다고 판단할 수 있다. (이진검색과 비슷한 원리)

5. 이진검색트리에 값을 **등록**하기: 루트 노드에서 시작하여 해당 노드의 값과 추가하고자 하는 값을 비교하여 왼쪽 자식 노드 혹은 오른쪽 자식 노드로 내려간다. 추가하고자 하는 값은 또 다른 리프 노드로써 이진검색트리에 연결된다.



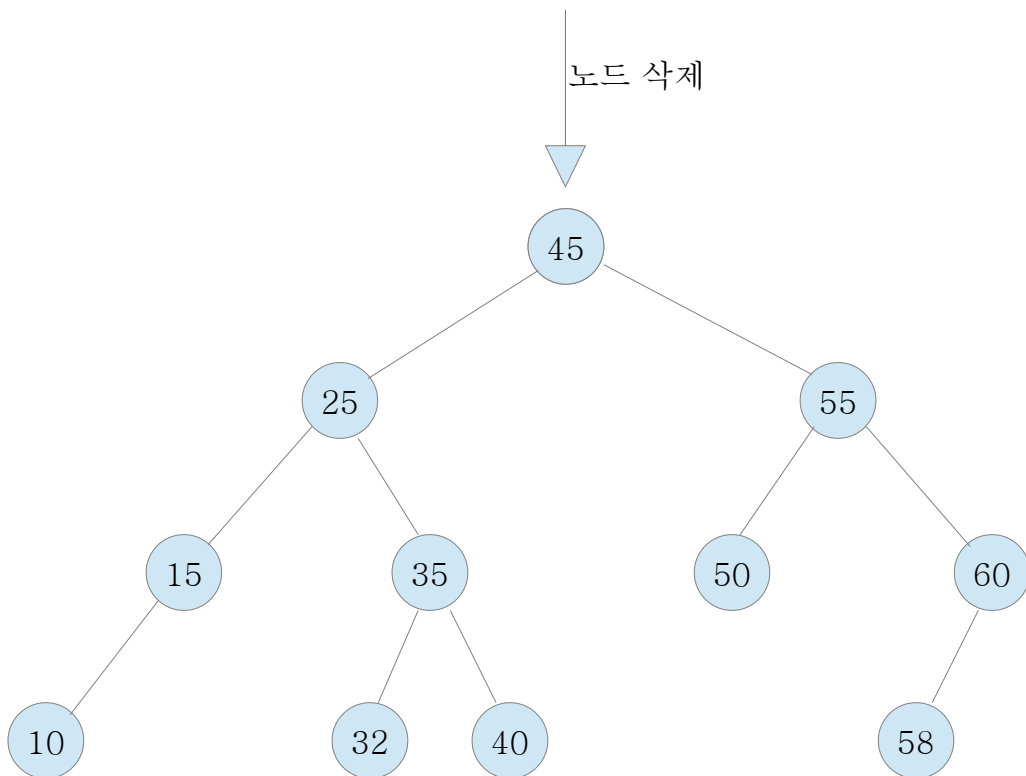
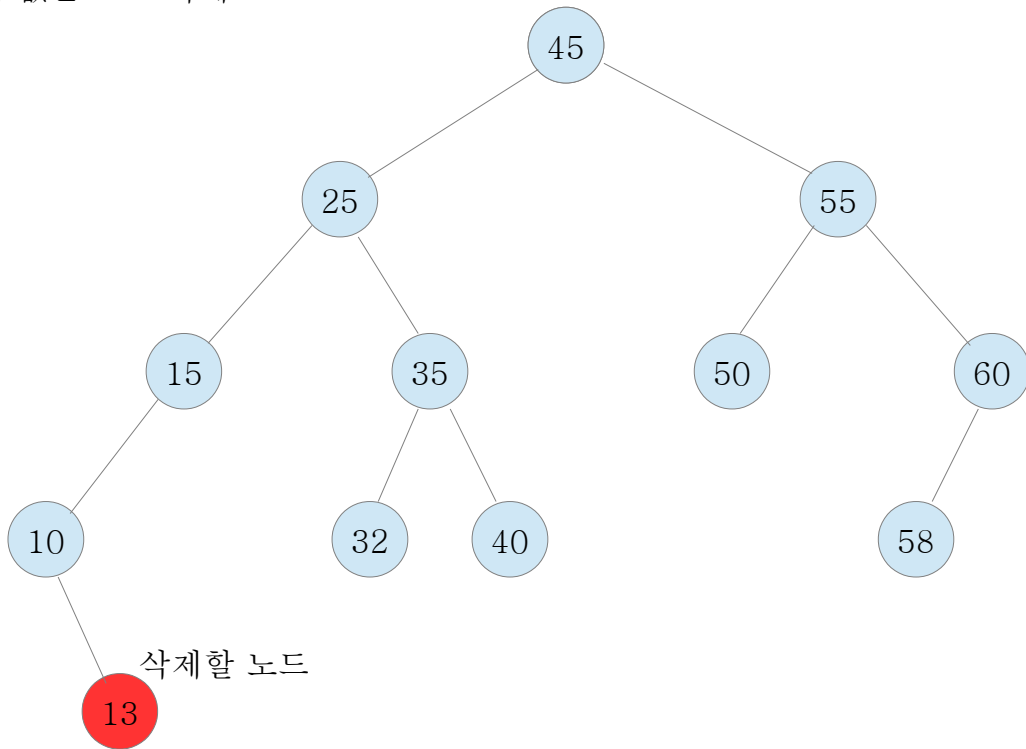
그리기 4: 이진검색트리에 값을 등록하는 예시

이 예시에서 추가할 노드 60은 비교하는 대상 노드 45 보다 크므로 오른쪽으로 내려간다. 추가할 노드 60은 오른쪽 자식 노드 55 보다 크다. 그러므로 추가할 노드 60은 55의 오른쪽 자식 노드가 되어 이진검색트리에 추가된다.

만약 52를 추가한다면 노드 50의 오른쪽 자식 노드로 이진검색트리에 연결된다.

6. 이진검색트리에서 값을 삭제하기

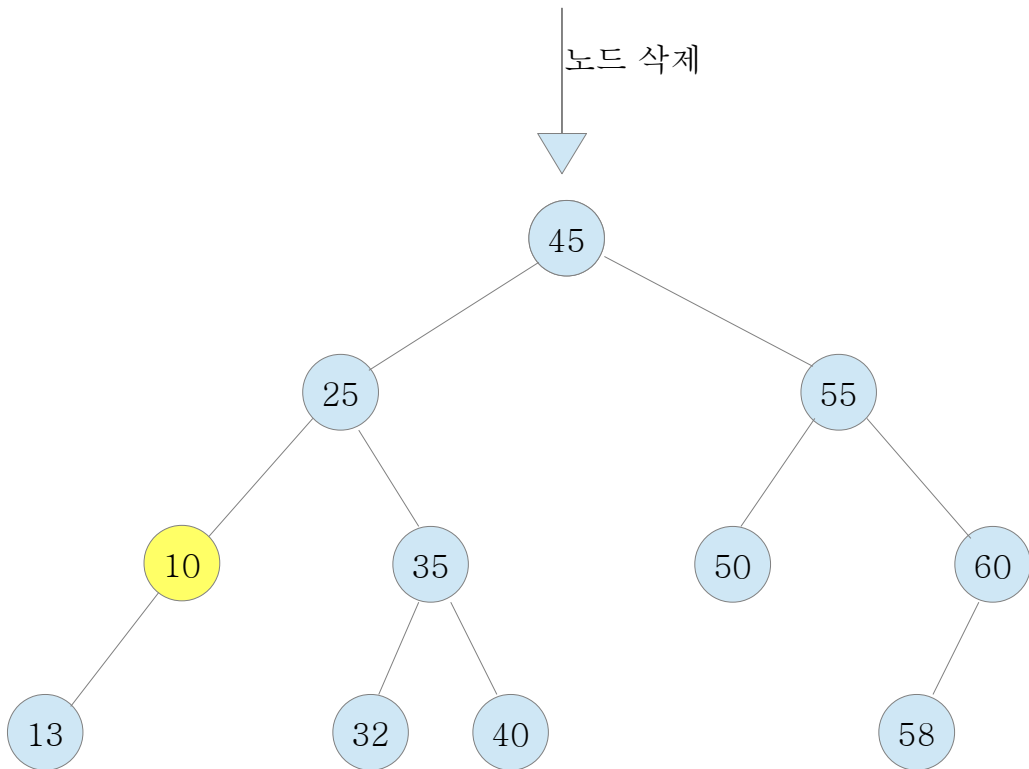
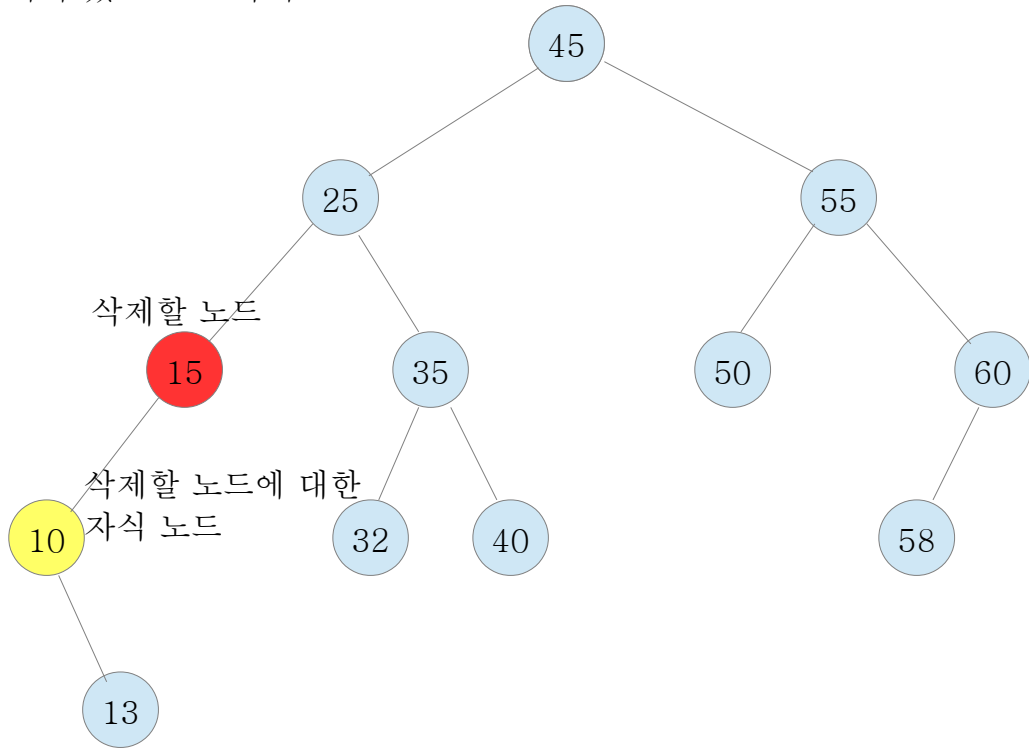
6.1. 자식이 없는 노드 삭제



그리기 5: 자식이 없는 노드 삭제 예시

자식이 없는 노드를 삭제할 때는 해당 노드를 바로 삭제하면 된다.

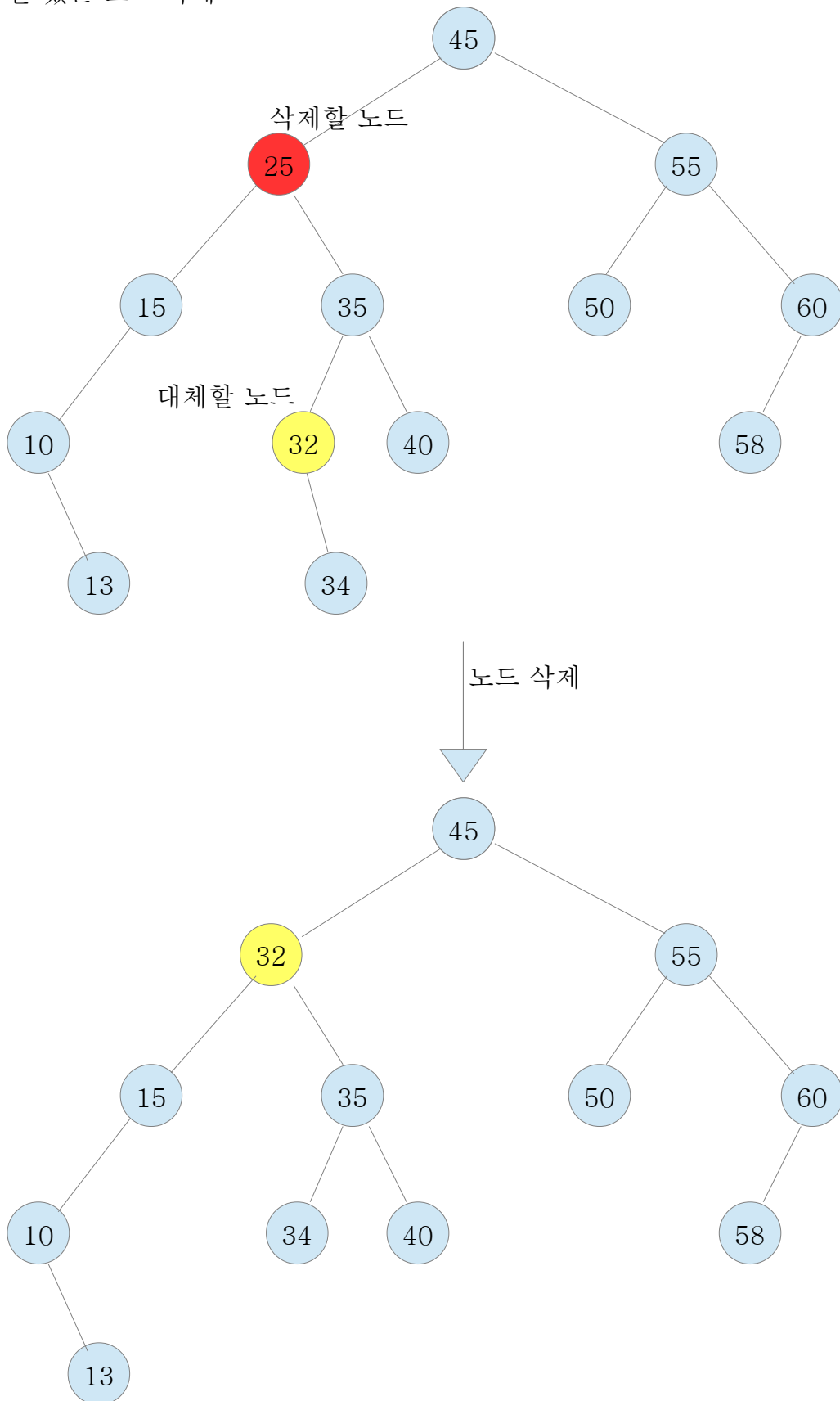
6.2 자식이 하나 있는 노드 삭제



그리기 6: 자식이 하나 있는 노드 삭제 예시

자식이 하나 있는 노드를 삭제할 때는 [삭제할 노드에 대한 자식 노드의 서브 트리]를 삭제할 노드로 끌어 올려준다.

6.3 자식이 둘 있는 노드 삭제



그리기 7: 자식이 둘 있는 노드 삭제 예시

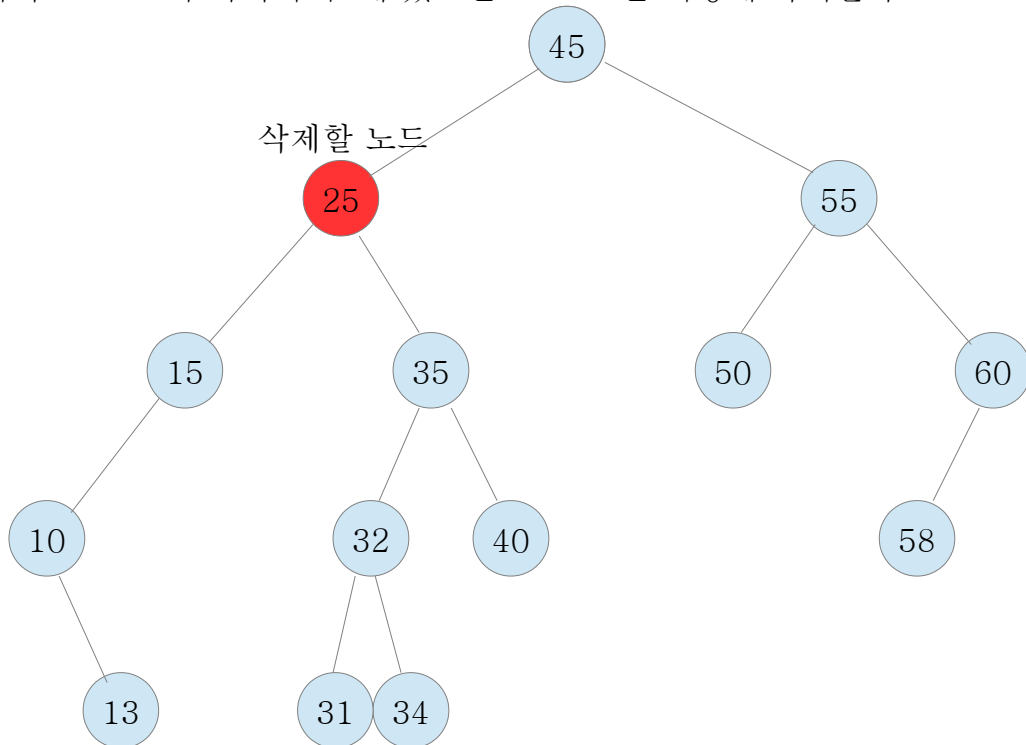
자식이 둘 있는 노드를 삭제할 때는 [대체할 노드]를 찾아서 [대체할 노드]를 [삭제할 노드] 자리에 위치 시킨다. (만약 [대체할 노드]에 오른쪽 자식 노드가 있는 경우 그 오른쪽 자식 노드를 [대체할 노드의 본래 자리]에 위치 시킨다.)

삭제할 노드에 대해 대체할 노드는 다음과 같은 기준으로 정한다: 삭제할 노드의 서브 트리에 속

한 노드 중의 하나로, 삭제할 노드보다 큰 노드 중 가장 작은 노드

삭제할 노드에 대해 대체할 노드는 다음과 같은 방법으로 찾는다: 삭제할 노드에서 우선 오른쪽 자식을 선택하고 더 이상 왼쪽 자식 노드가 없을 때 까지 왼쪽 자식 노드를 선택한다. 대체할 노드는 리프 노드가 될 수도, 리프 노드가 안 될 수도 있다. (오른쪽 자식 노드만 있고 왼쪽 자식 노드가 없는 노드 역시 대체할 노드가 될 수 있다.)

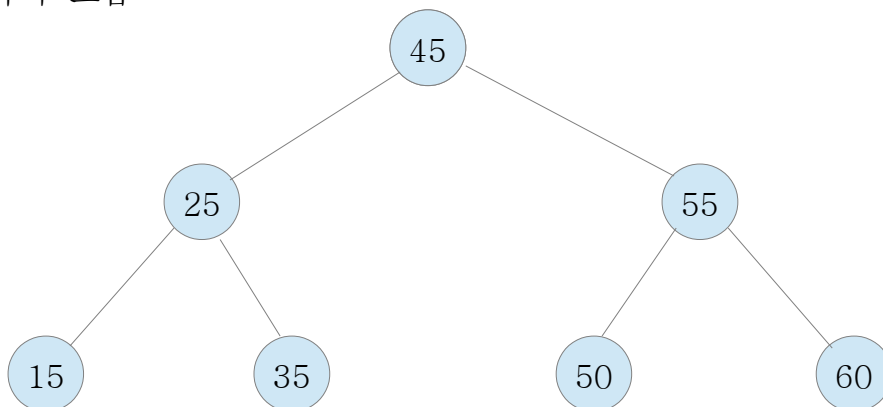
cf. 위 예시에서 노드 32가 자식이 두 개 있으면 노드 25를 어떻게 삭제할까?



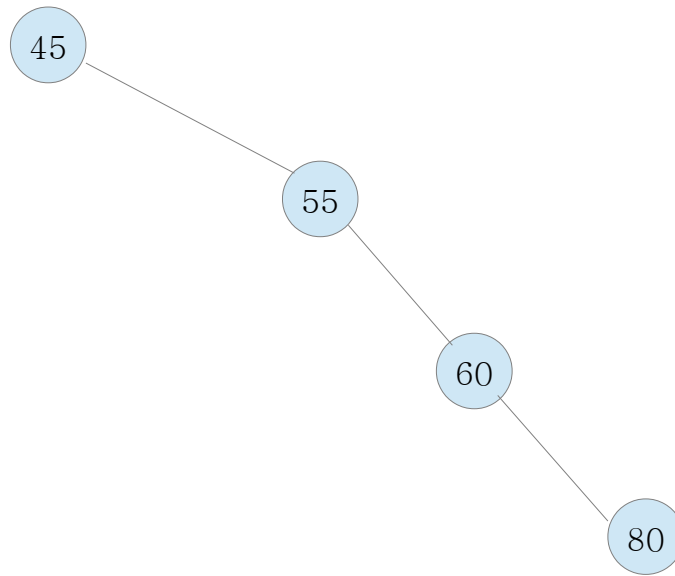
그리기 8: 자식이 둘 있는 노드 삭제 예시

이 경우 사실 노드 32는 대체할 노드가 아닌 것이다. 이 경우 노드 31이 대체할 노드가 되어 노드 25의 위치로 이동한다.

7. 이진탐색트리의 두 모습



이진탐색트리에 값을 넣을 때 크고 작은 값이 잘 섞여서 들어오면 각 노드들이 가능한 두 자식 노드들을 가지게 된다. 이 경우 트리의 높이는 $\log N$ 에 비례하게 된다.



이진탐색트리에 값을 넣을 때 값이 오름차순, 내림차순으로 들어오면 해당 트리는 일직선 모양을 가지게 된다. 이 경우 트리의 높이는 N 에 비례하게 된다.

8. 시간복잡도

이진탐색트리의 insert, erase, find, update의 시간복잡도는 $O(\log N)$ 이다. 이 연산들을 상수시간에 수행할 수 있는 해시 테이블에 비해 시간복잡도 상 불리하지만, 이진탐색트리는 값이 정렬되어 들어간다는 특징이 있다.