Ch14. Binary Search

BST

빠른 조회, 삽입 및 제거 작업을 용이하게 하는 데이터 구조

왜 조회, 삽입, 제거가 빠른가?

-->한 쪽을 선택하면 다른 쪽 가능성을 상실하므로 인해 탐색할 영역이 줄어들기 때문에

평균 시간 복잡도 : O(logN)

BST rules

왼쪽 child의 값은 parent의 값보다 작아야 합니다.

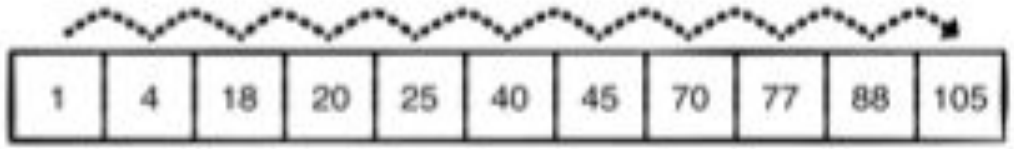
오른쪽 child의 값은 parent의 값보다 크거나 같아야 합니다.

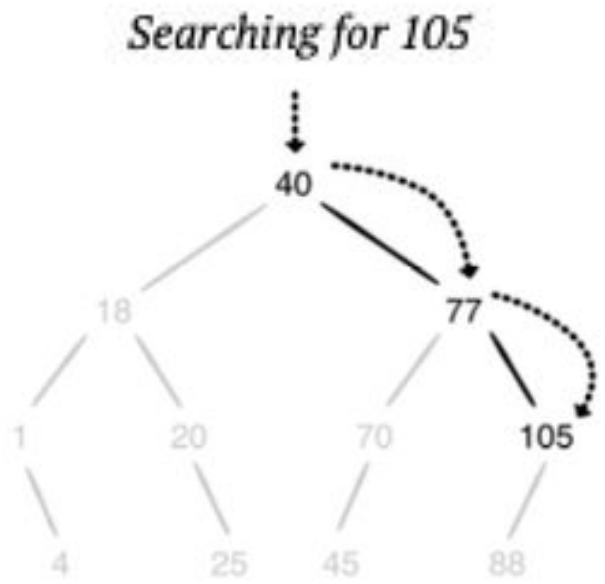
Array vs BST

In look-up case

array : O(n)

BST : O(log n)





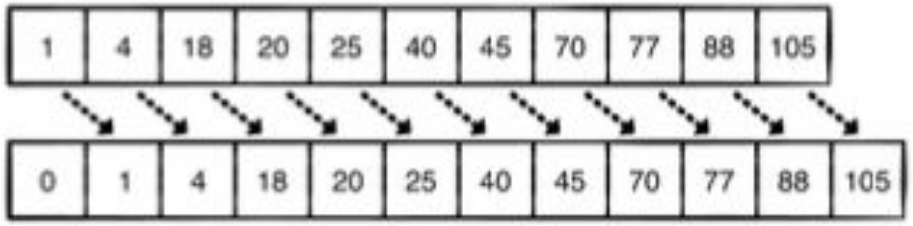
Searching for 105

Array vs BST

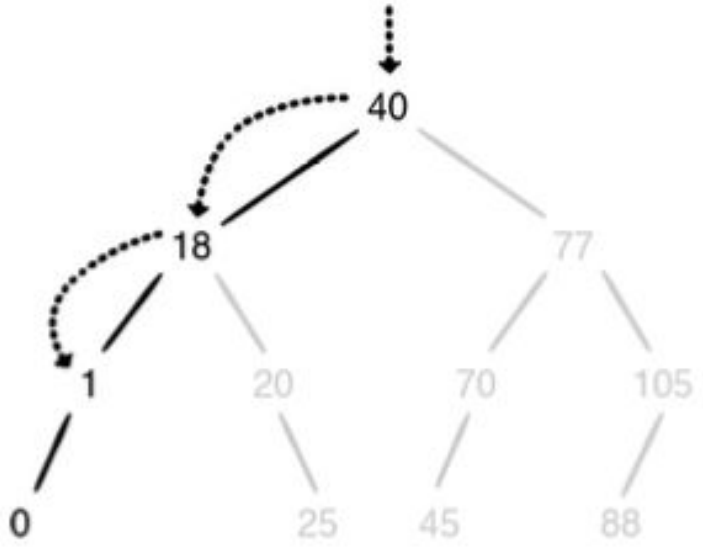
In Insertion case

array : O(n)

BST : O(log n)



Inserting 0 in sorted order

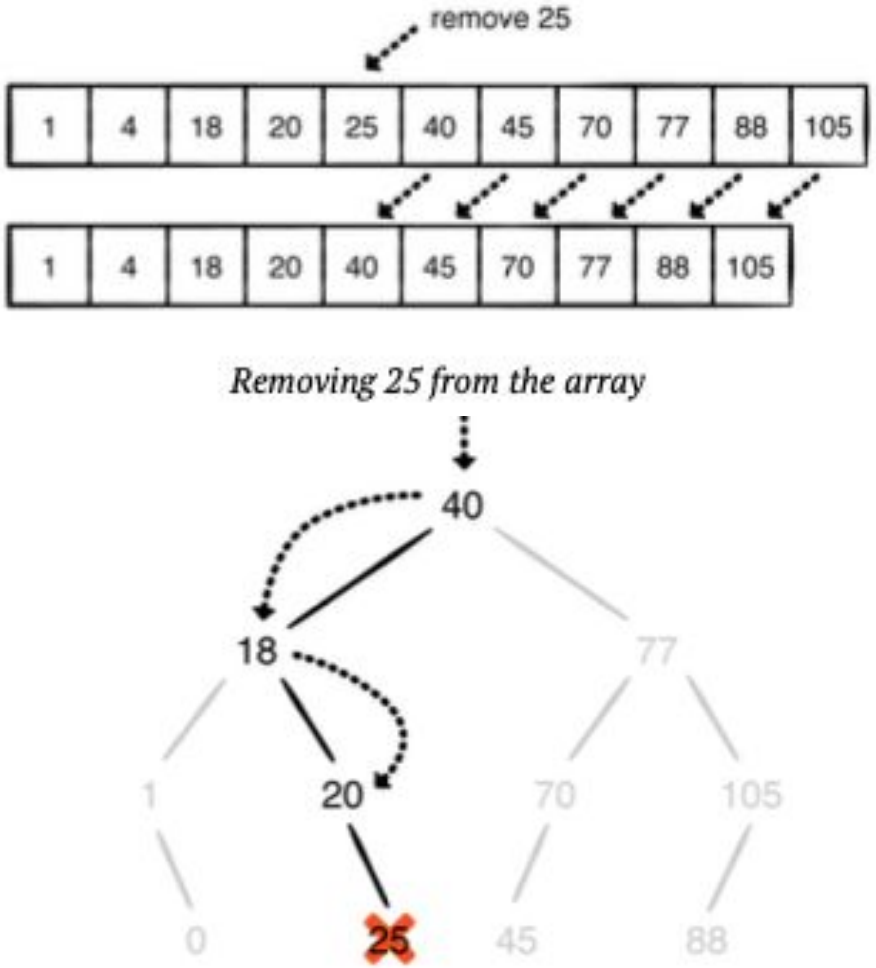


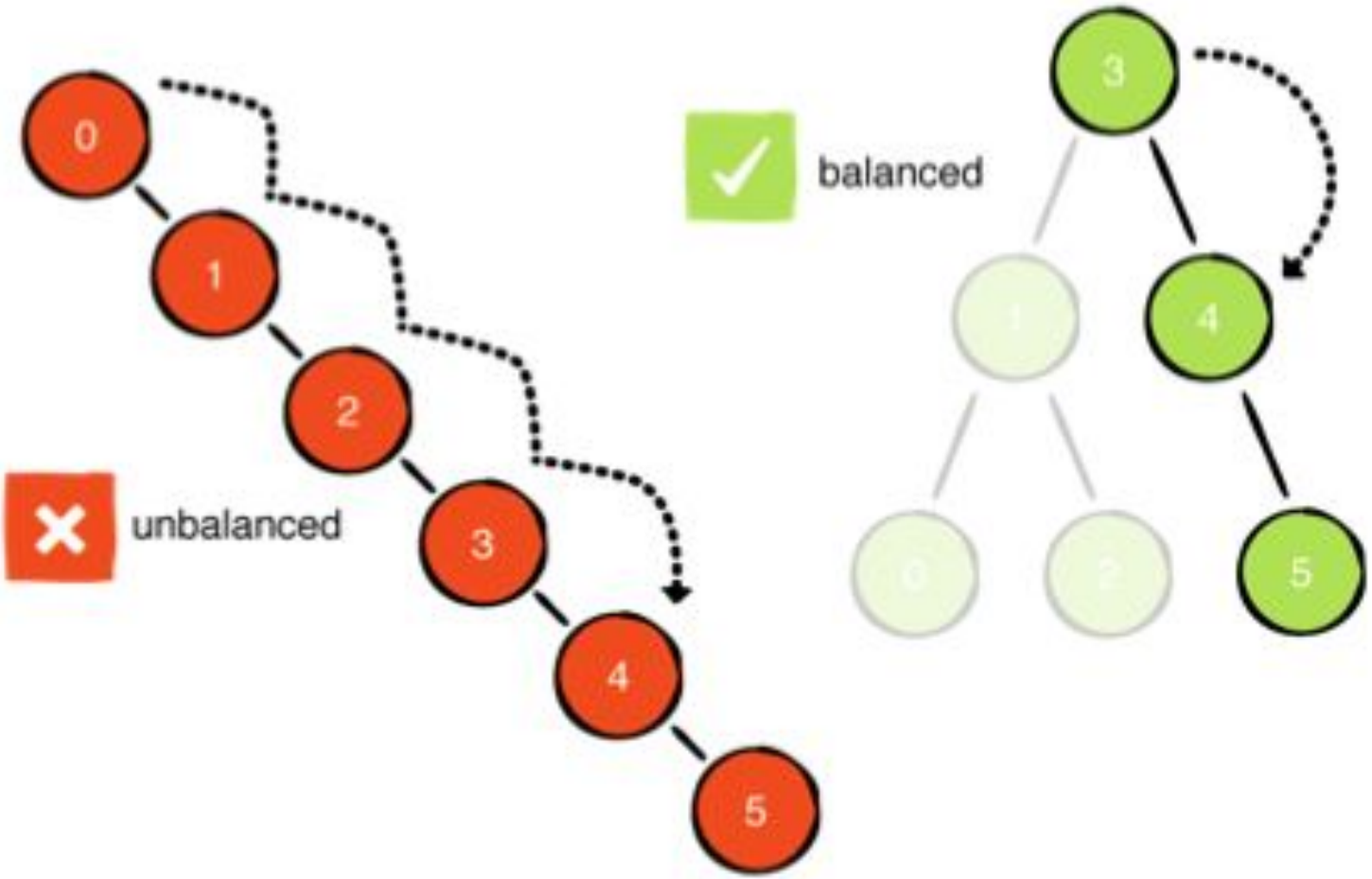
Array vs BST

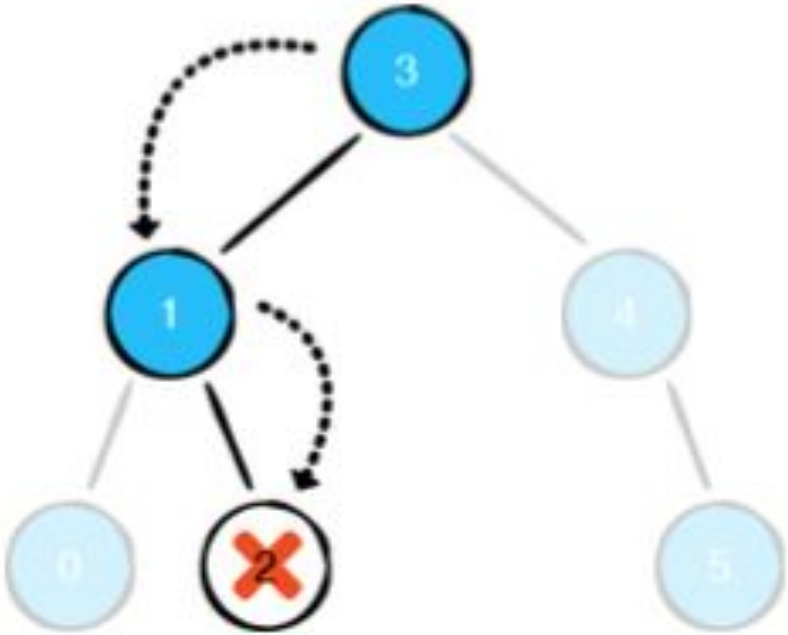
In Removal case

array : O(n)

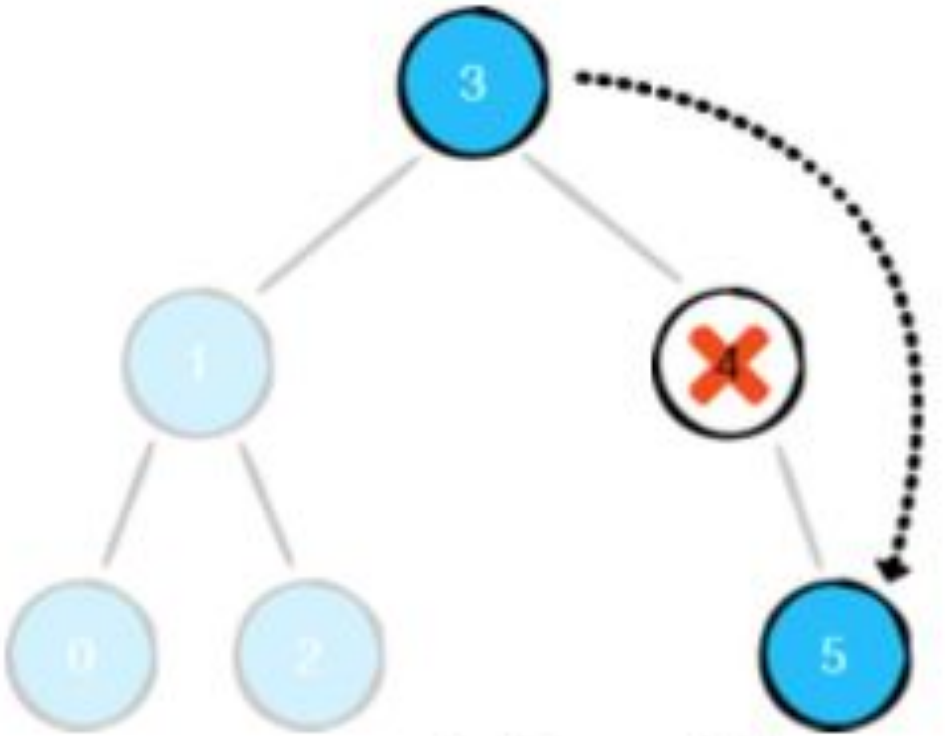
BST : O(log n)



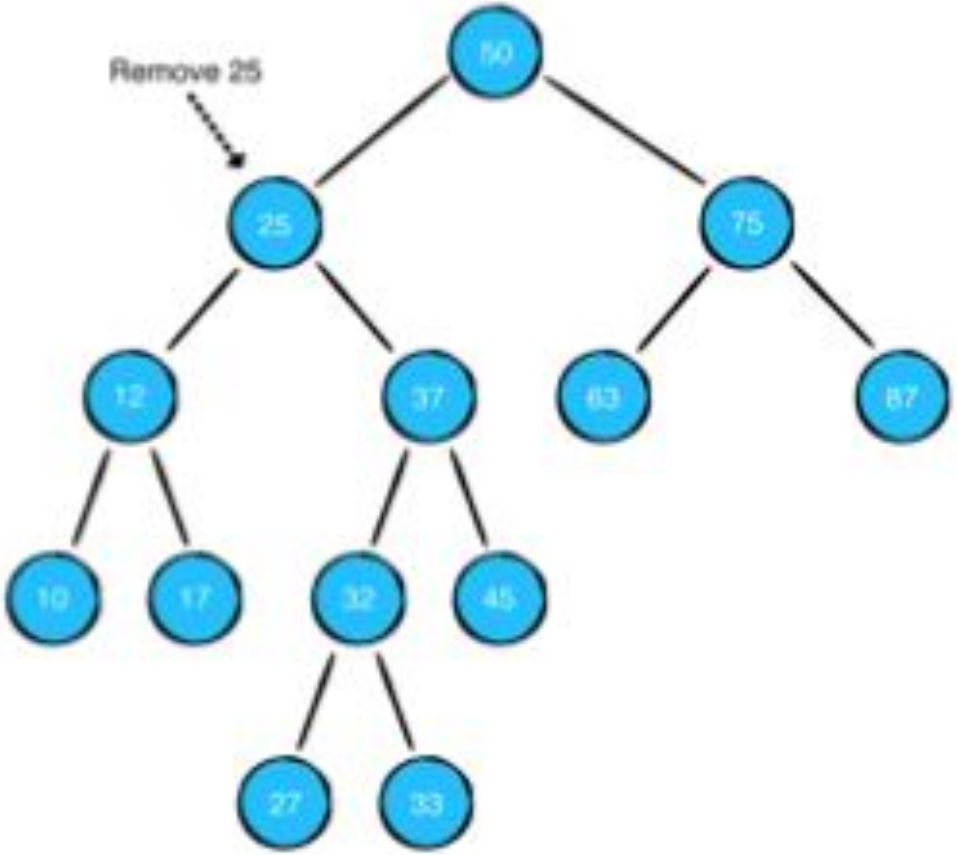

0
1
2
3
4
5
unbalanced
balanced
3
1
4
0
2
5

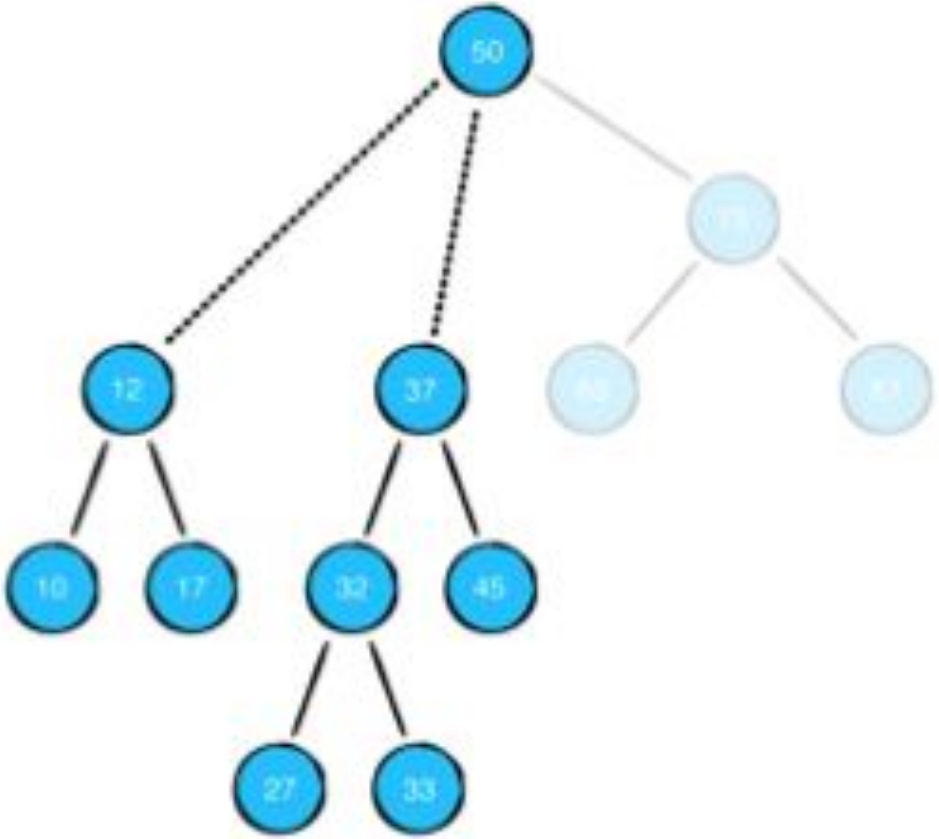


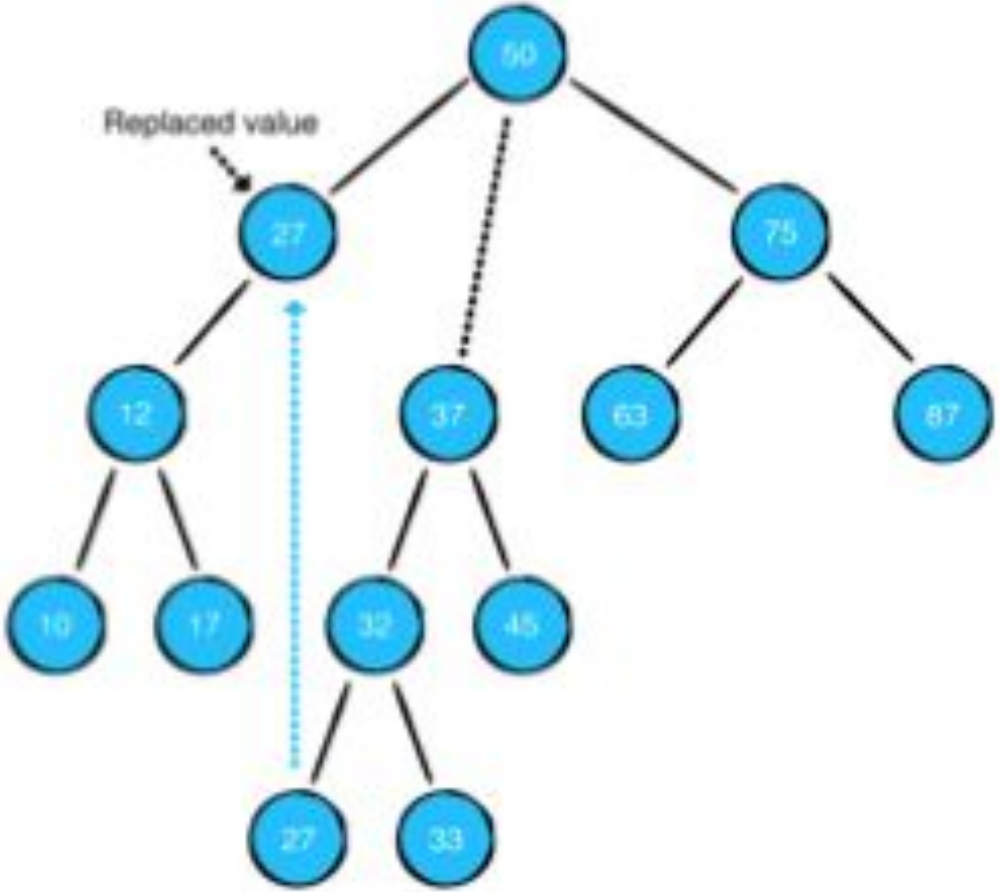
removing 2

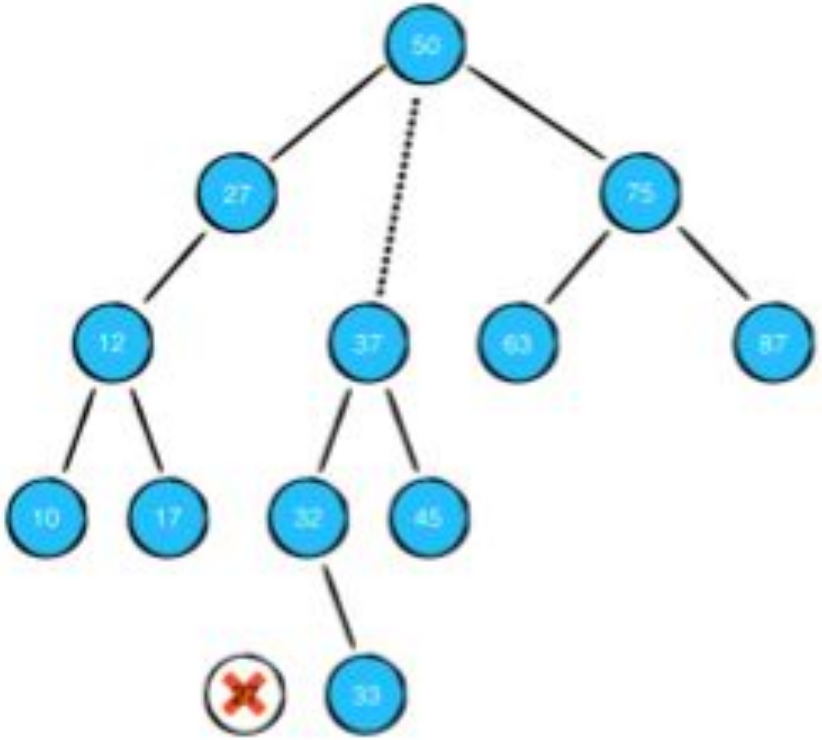


removing 4, which has 1 child


Remove 25
50
25
75
12
37
63
87
10
17
32
45
27
33


50
12
37
10
17
32
45
27
33


50
Replaced value
27
75
12
37
63
87
10
17
32
45
27
33


50
27
75
12
37
63
87
10
17
32
45
33