



EDDI

Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv2 과정

제 1기

2021. 10. 25

손표훈

CONTENTS

- Binary Tree 삭제 구현(자식노드 2개 일 때)
- AVL Tree

Binary Tree 삭제 구현(자식 노드 2개 일 때)

※ Binary Tree 삭제 구현(자식 노드가 2개 있을 때)

→ 삭제하고자 하는 노드를 찾는다

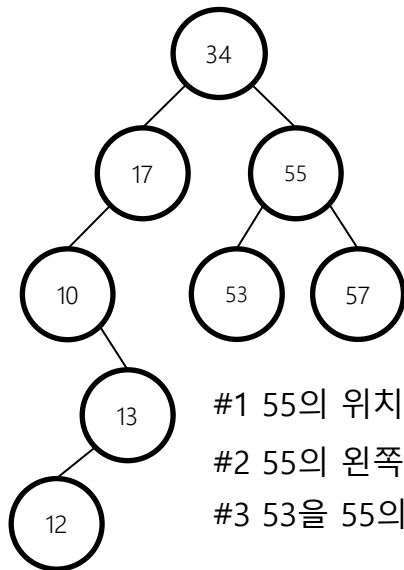
→ 삭제 방법에 2가지가 있다

(1) 해당 노드의 왼쪽 자식노드에서 가장 큰 값을 삭제하는 노드에 대입

(2) 해당 노드의 오른쪽 자식노드에서 가장 작은 값을 삭제하는 노드에 대입

→ 최대, 최소값을 가지고 있던 노드의 메모리 해제

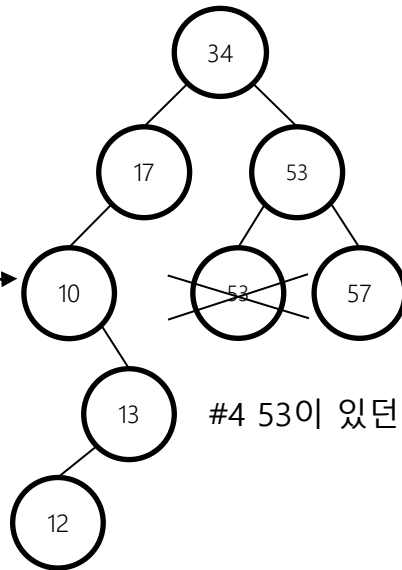
※ 예시 : 55를 지운다고 했을 때(왼쪽 최대 값을 찾을 때)



#1 55의 위치를 찾는다

#2 55의 왼쪽 자식 노드 중 최대 값을 찾는다

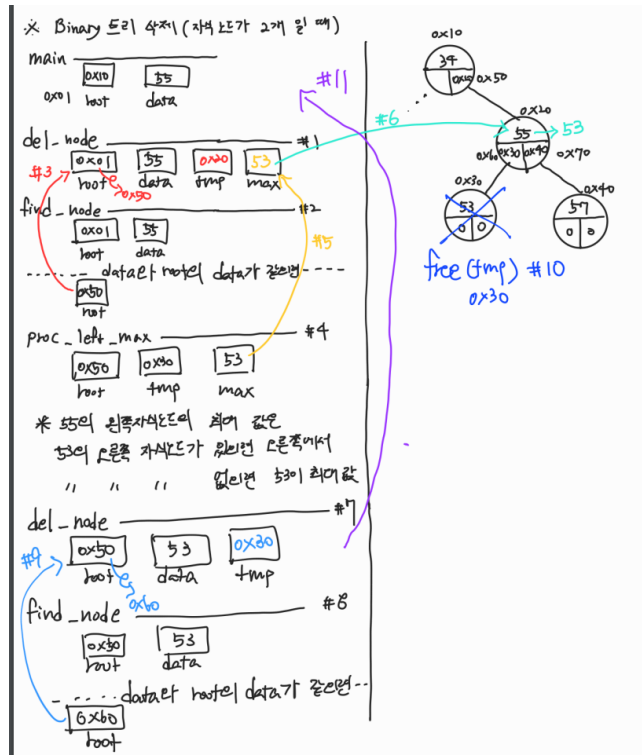
#3 53을 55의 노드에 대입



#4 53이 있던 노드의 메모리 해제

Binary Tree 삭제 구현(자식 노드 2개 일 때)

※ Binary Tree 삭제 구현(자식 노드가 2개 있을 때)



AVL Tree

※ AVL Tree의 특징

- 왼쪽, 오른쪽 서브트리의 높이가 최대 1인 트리
- 1이 초과되면 1이 되게 균형을잡는다
- AVL트리의 시간복잡도는 $\log N$ 이 된다(N은 노드의 개수): 이진트리는 왼쪽, 오른쪽 둘 중 하나에서 데이터를 찾기 때문

ex) 20, 15, 3, 12, 5, 11, 6을 insert 할 때

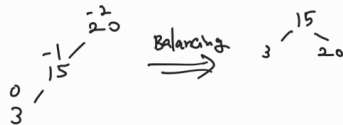
#1. 20 insert



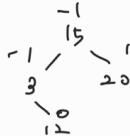
#2. 15 insert



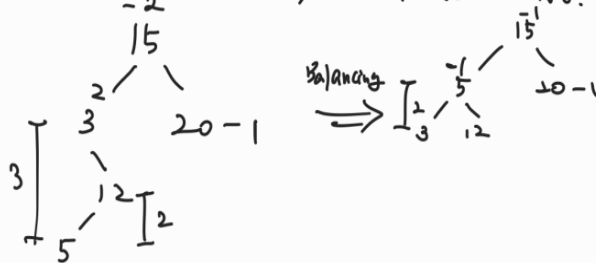
#3. 3 insert = 왼쪽노드와 오른쪽노드의 높이가 2이므로 높이가 높은 왼쪽에서 Balancing 수행, 왼쪽 레이어 중 가장 큰 값을 루트노드에 대입



#4. 12 insert



#5. 5 insert : 5 삽입시 3노드에서 높이차 발생 5에는 왼쪽보다 큰 쪽이 더 높으므로 오른쪽노드에서 리스 값을 3에 재삽입.



#6. 11 insert : 11 삽입시 5노드에서 2차이 Balancing 후에도 15에서 2차이 Balancing 1번 더 수행.

