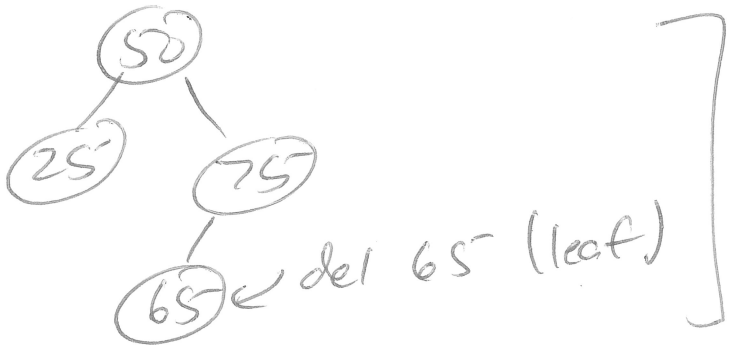
COP 3502 11/1/2021

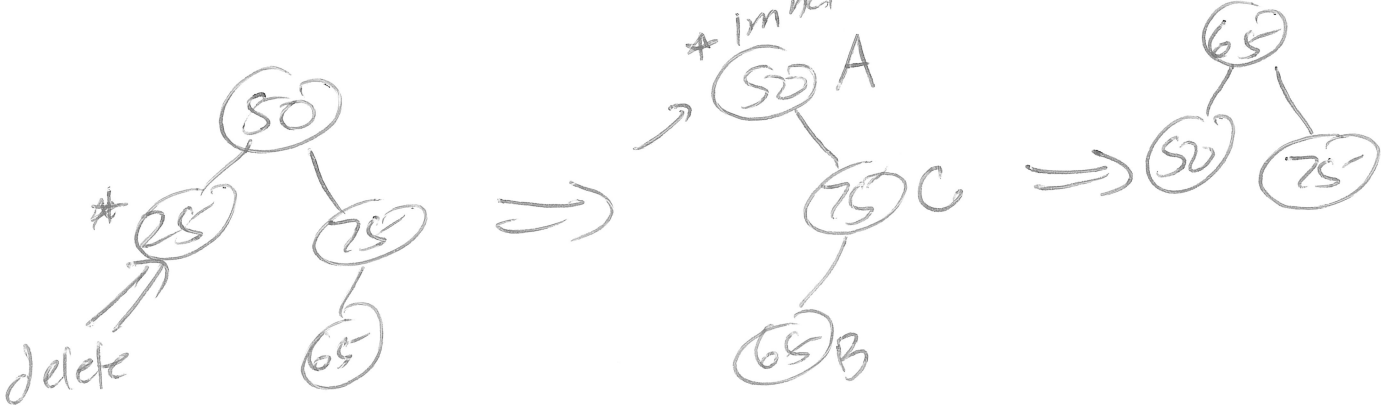
(1) AVL Tree Delete
(2) AVL Tree Code

Cases Delete



Many cases, when no new imbalance is introduced - same as regular BST delete.

Note: We never physically delete a node with 2 children!

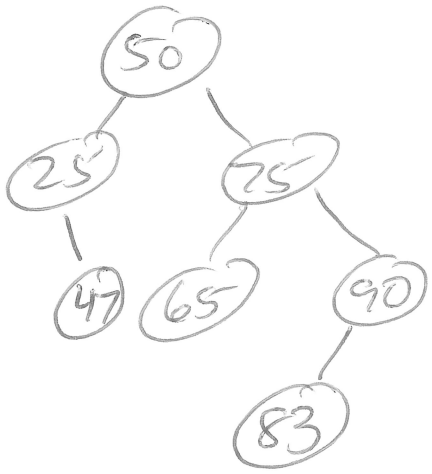


As you trace up the ancestral path from the deleted node, fixing one imbalance, may cause a new one further up the tree.

Delete Run time

1) Go down tree $O(\lg n)$
+ 2) Go back up tree $O(\lg n)$
+ 3) Rebalances at most $O(\lg n)$, each constant time

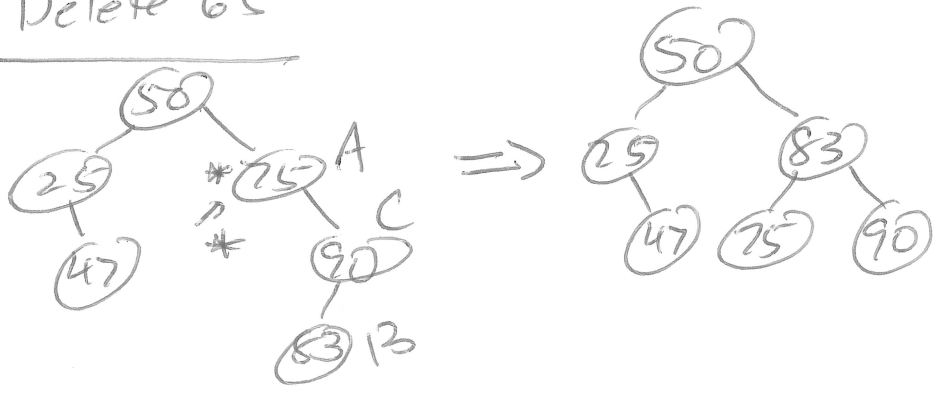
$O(\lg n)$



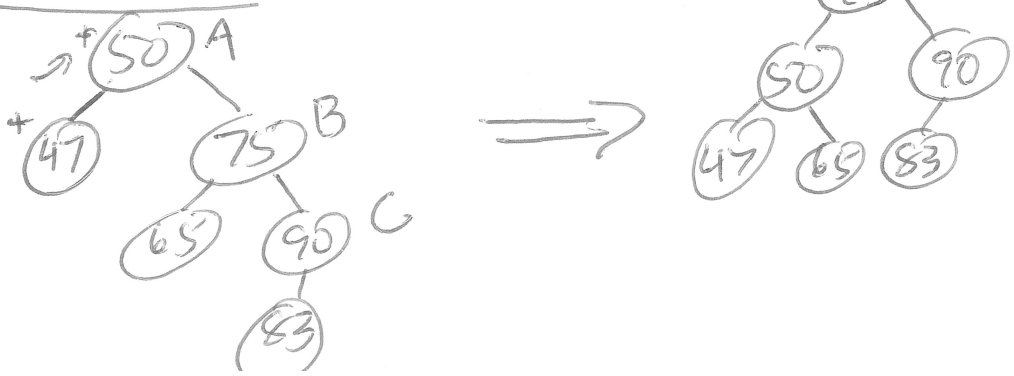
(1) Delete 65

(2) Delete 25

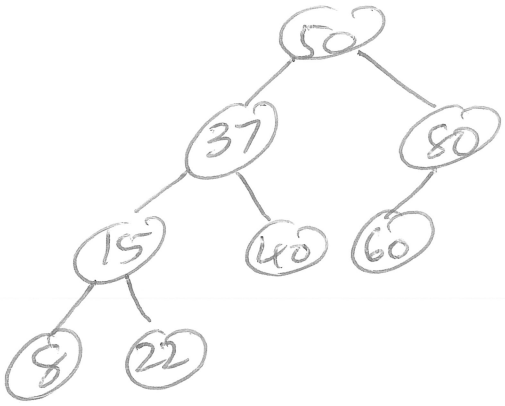
Delete 65



Delete 25



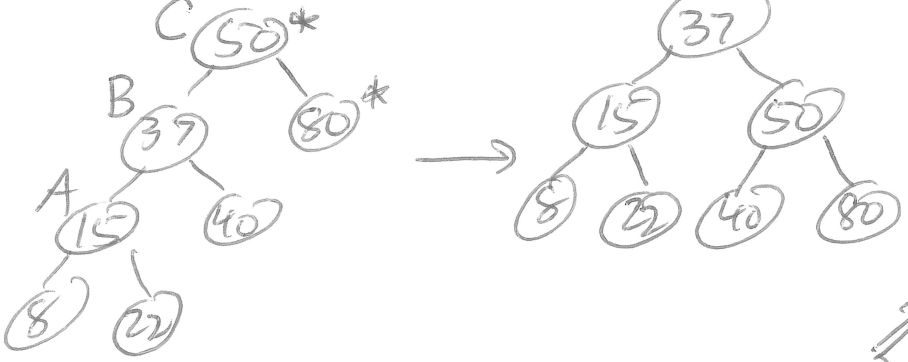
Delete



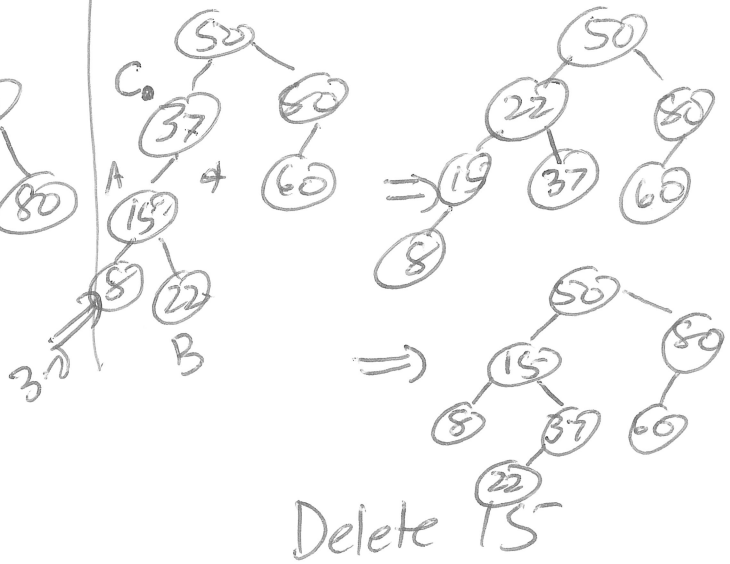
(1) Delete 60

(2) Delete 40

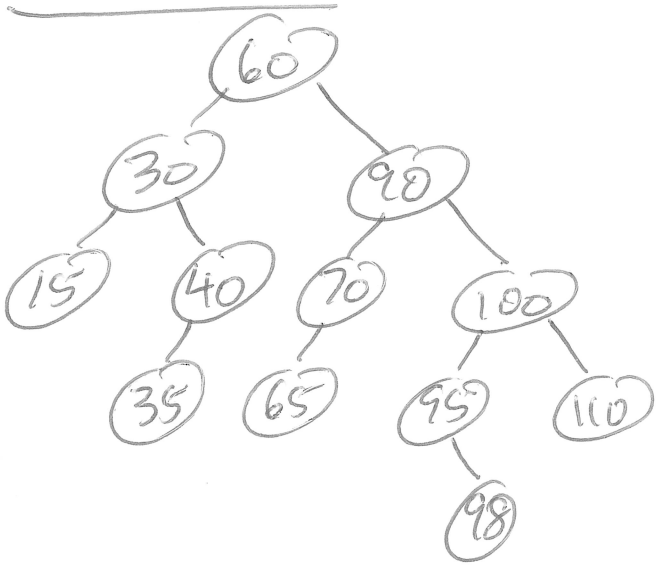
Delete 60



Delete 40



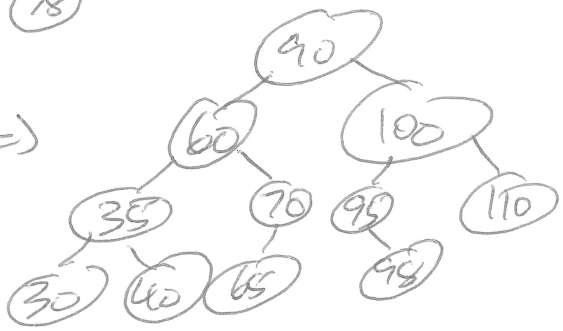
Delete 15



Step 1

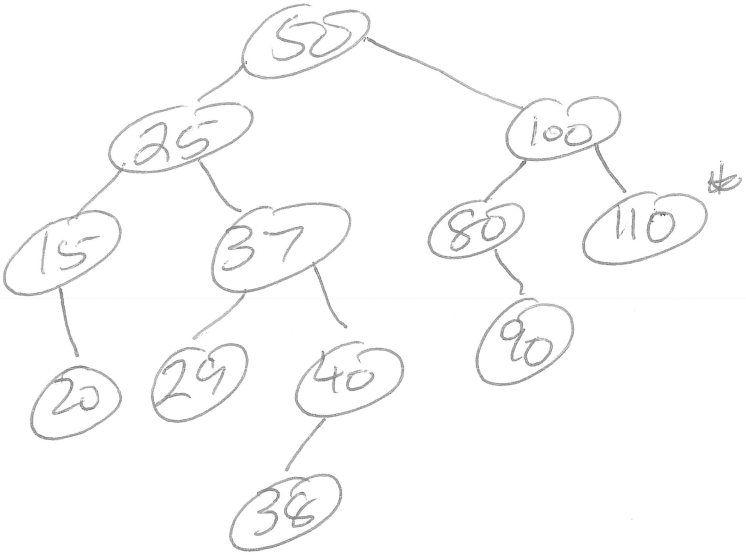

60
A
30
C
90
40
70
100
B
35
65
95
110
98
A
60
V
35
B
90
C
30
40
70
100
65
95
110
98

Step 2 =>

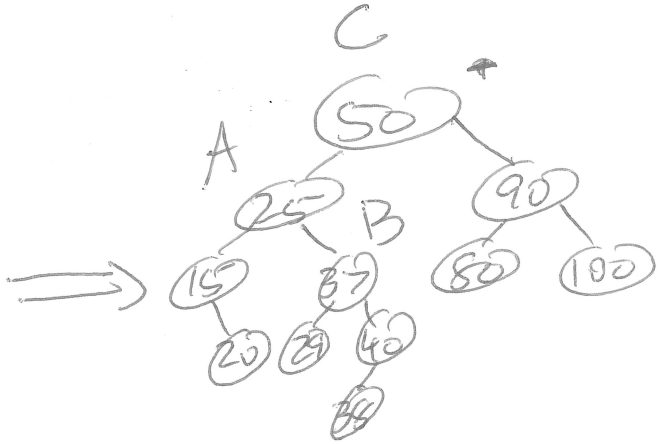

90
60
100
35
70
95
110
30
40
65
98

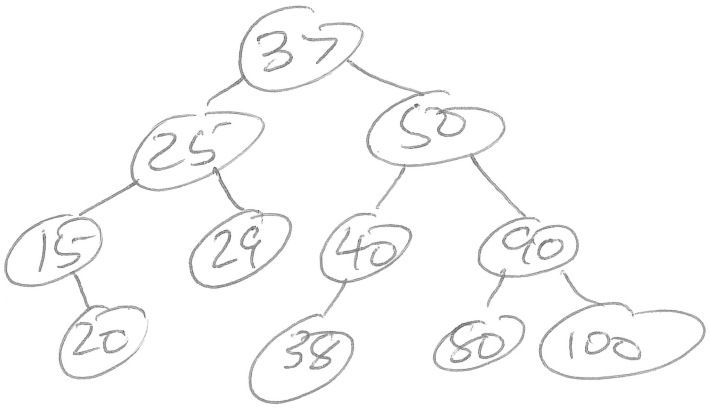
Delete

Delete 110

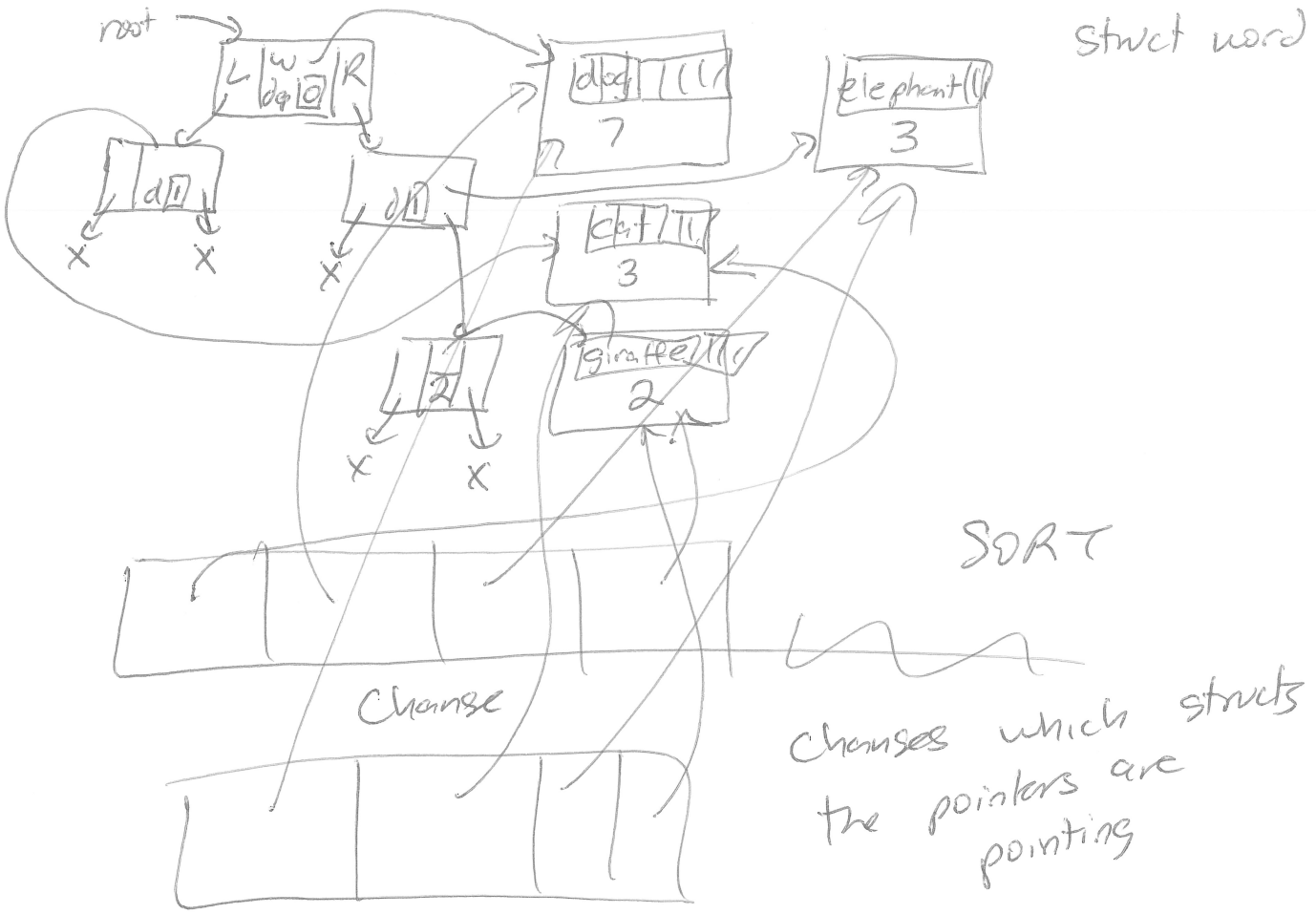


Step 1





PS suggested struct picture



struct word

changes which structs the pointers are pointing

compare(word* w1, word* w2) {

}