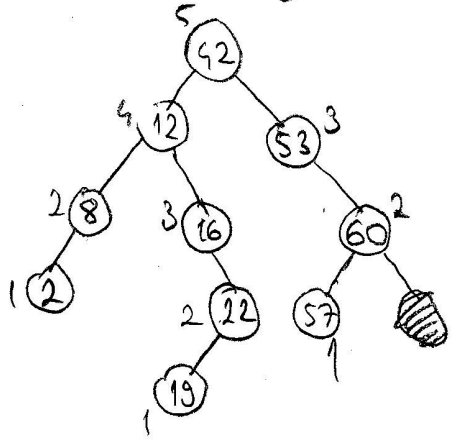
Homework 07 - Ceng 310 Onur Yaman

1. Binary Search Trees

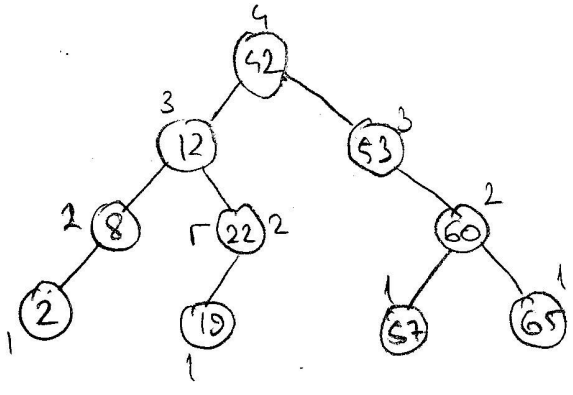
Task-1

- After removing 65 from the tree



Since 65 is leaf node, do nothing.

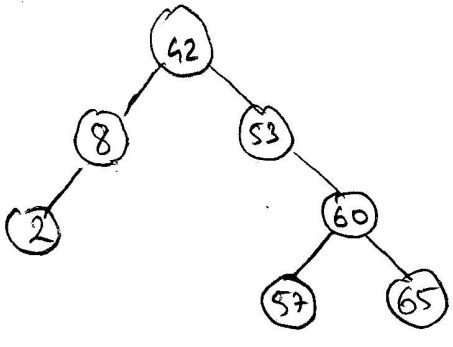
- After removing node 16 from the orignal tree



Since 16 has one child r=22, this is simple case, 22 would 'rigth child of 12.

The tree is balanced, so rotation is not needed.

- After removing node 12 from the original tree



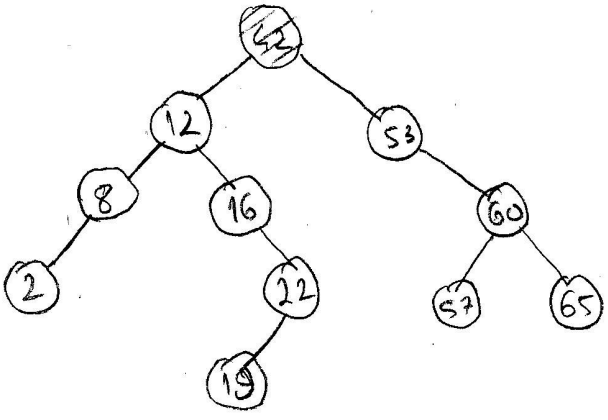
r = before(p), where p=12

Since 12 has two child, it should be replaced by r=8.

Then the state would be simple case.

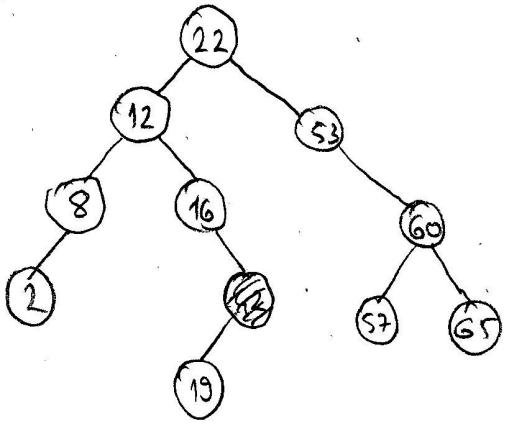
The tree is balanced now.

• After removing node 42 from the original tree



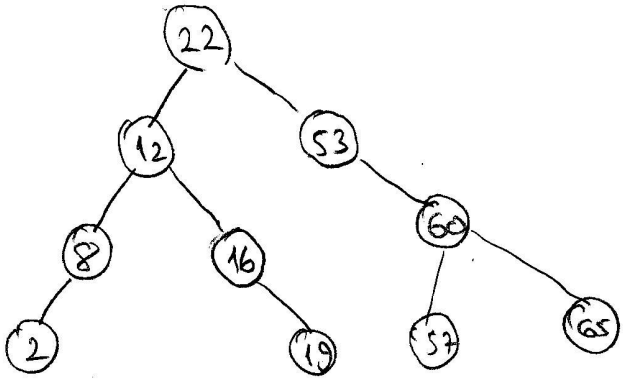
before(42) = 22

replace 22 with 22



delete 42

simple case



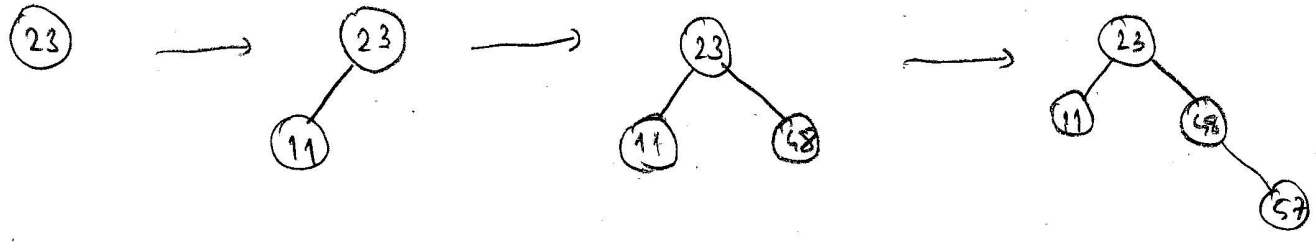
Task-2

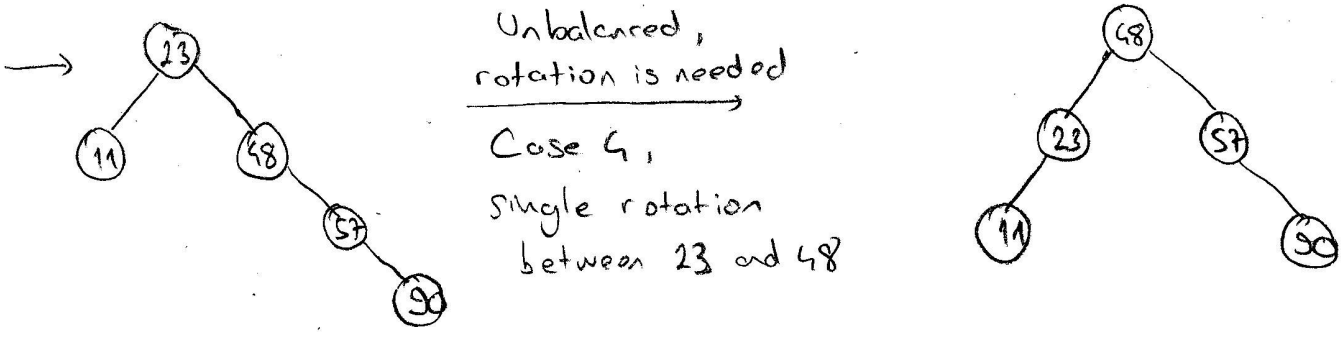
Pseudo code for the before()

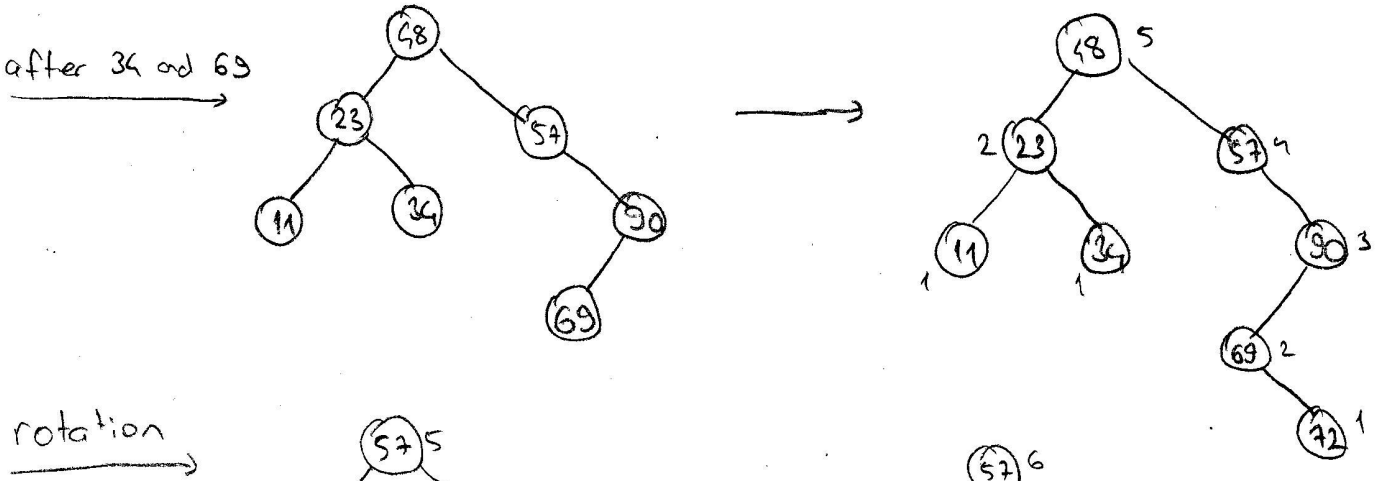
```
before(p):
    if left(p) is not None then
        walk = left(p)
        while rigth(walk) is not None do
            walk = rigth(walk)
        return walk
    else
        walk = p
        child = child(walk)
        while child is not None and walk == left(child) do
            walk = child
            child = child(walk)
        return child
```

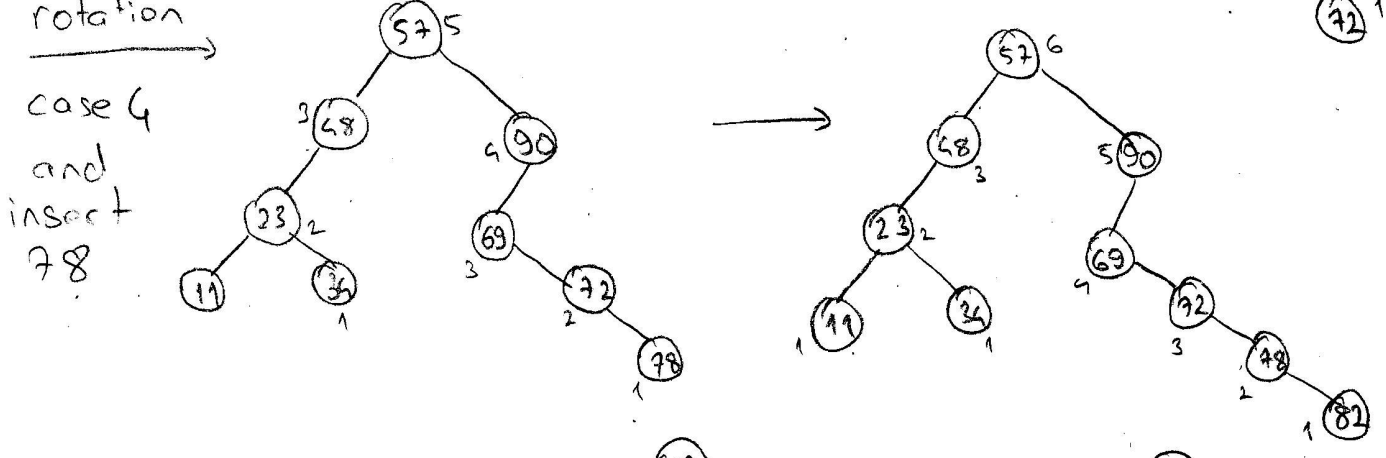
2. AVL Trees

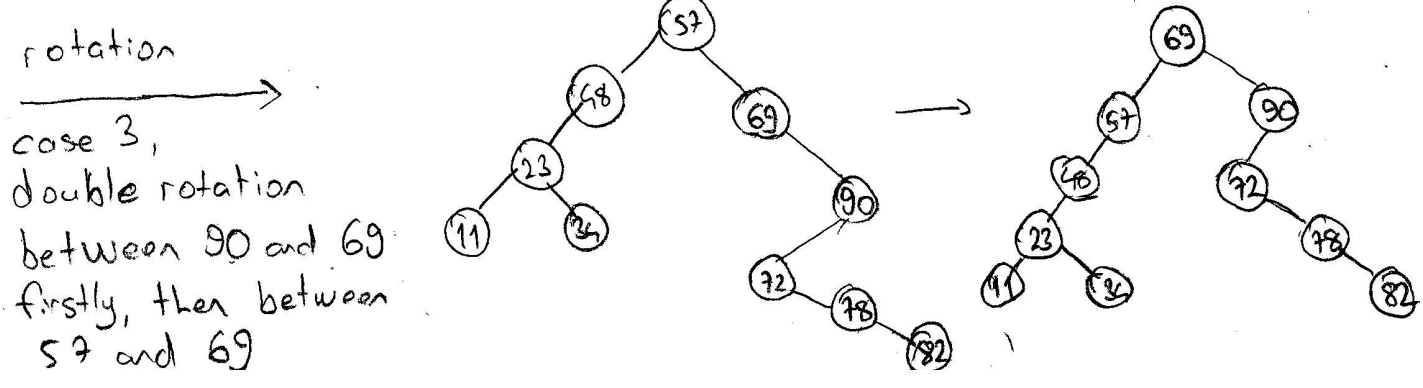
Task-3:



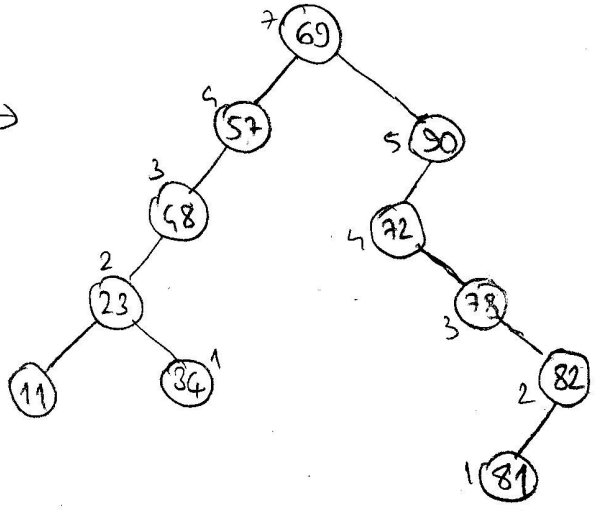








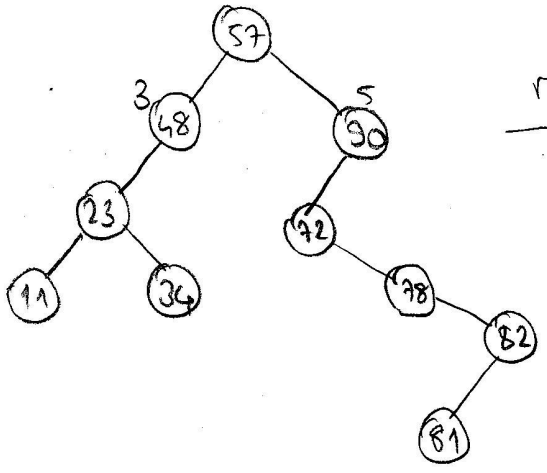
insert 81 →



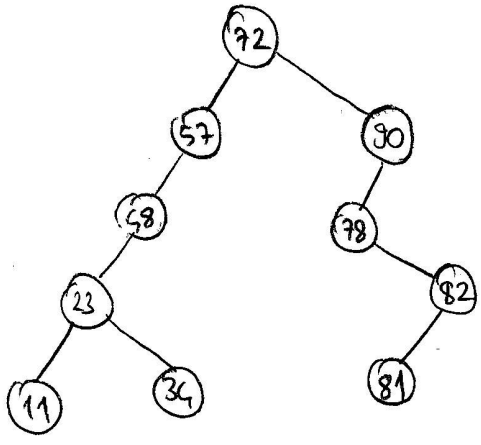
Task-4

Delete 69 from the tree above: note: before(69) = 57
after replacing 69 with 57, delete operation is applied.

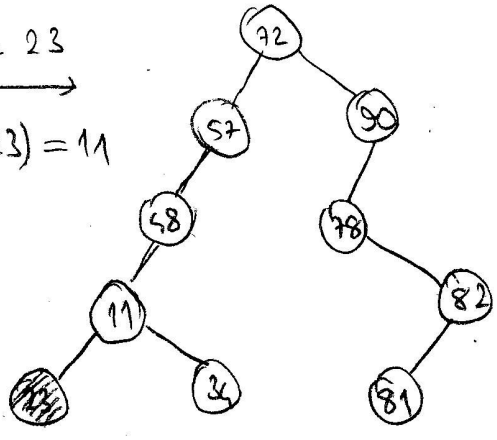
→



rotation →
case 3
double
rotation



delete 23 →
before(23) = 11



delete 11 →