Asuman Aydın

21502604

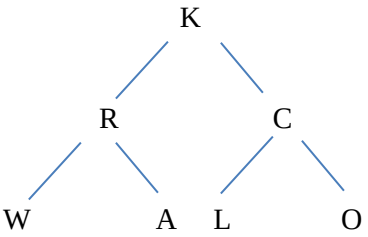
CS 202, Fall 2018

Homework #2 – Binary Search Trees

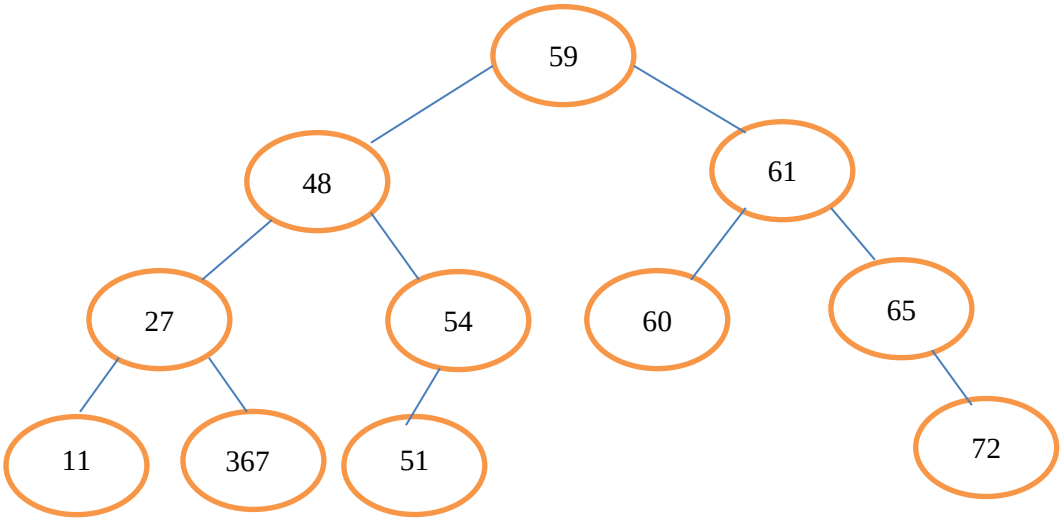
Question 1:

A- In Order: 3-5*8/4^1+7
Pre Order: /-3*58+^417
Post Order: 358*-4/^7+/

C- In Order: W R A K L C O

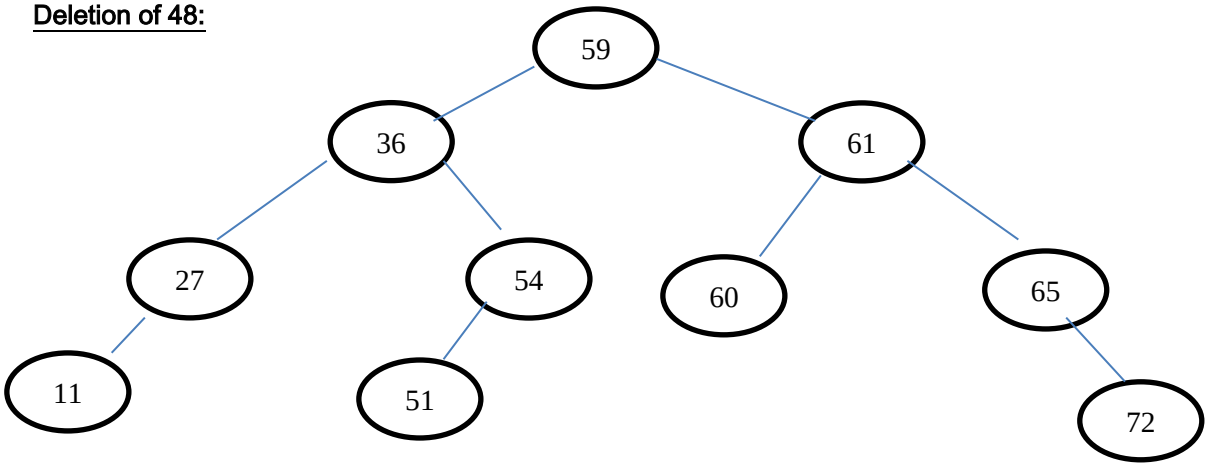


B-

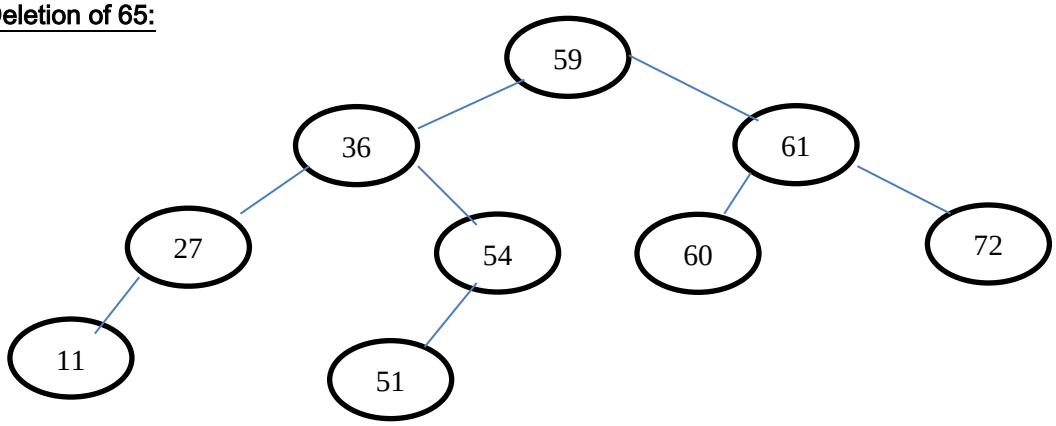


The final tree after all insertions.

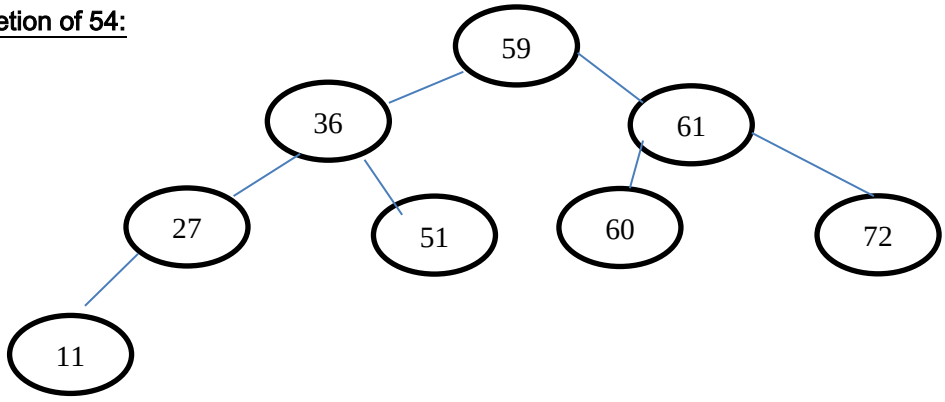
Deletion of 48:



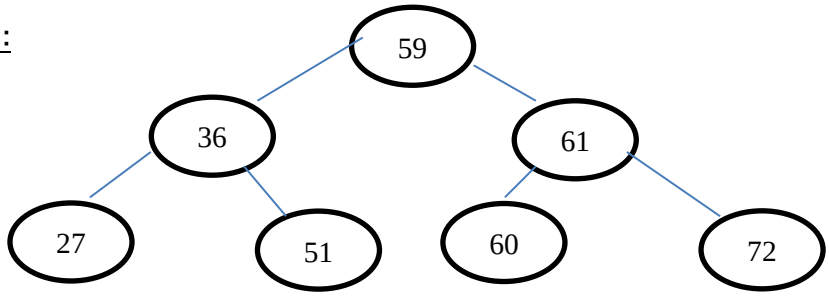
Deletion of 65:



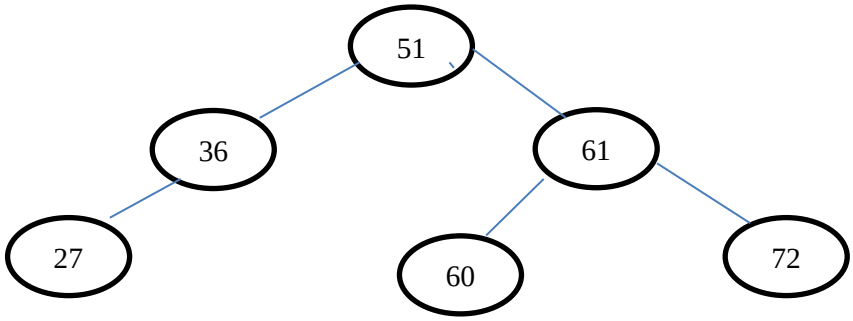
Deletion of 54:



Deletion of 11:



Deletion of 59:

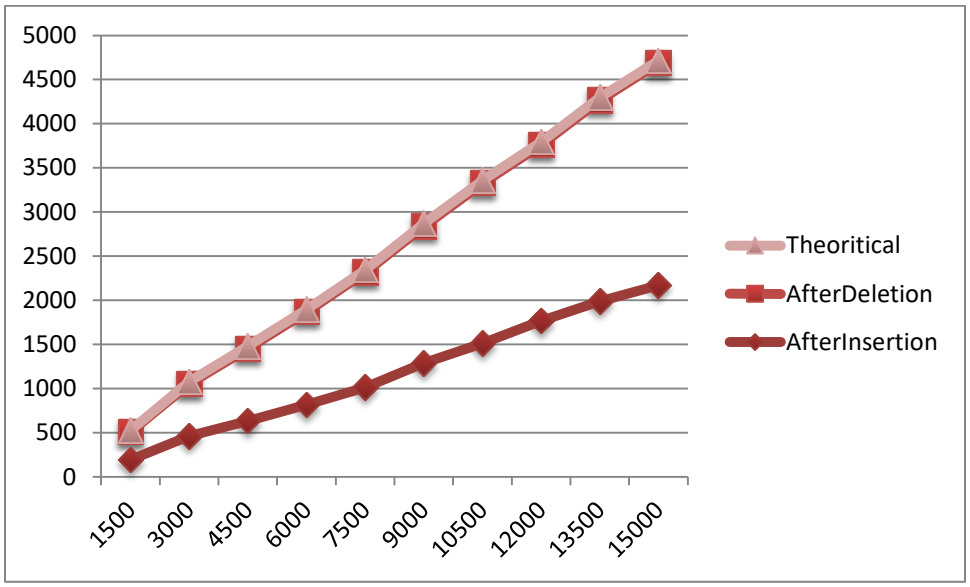


Question 3:

INSERTION

Tree Size	Time Elapsed
1500	193
3000	460
4500	633
6000	818
7500	1013
9000	1288
10500	1513
12000	1765
13500	1987

DELETION	
Tree Size	Time Elapsed
1500	314
3000	592
4500	819
6000	1050
7500	1300
9000	1549
10500	1812
12000	1996
13500	2277
15000	2513



ELAPSED TIME vs SIZE OF ARRAY

ANALYSIS:

- Theoretically, the time complexity depends on the height and for the worst case height would be same with the number of nodes. However, for this assignment, we insert random numbers and delete even more random numbers so for the theoretical calculation I took the height of every insertion. So it became in same portion of time elapsed in deletion.
- For the insertion of sorted array numbers, it does not matter whether it is ascending or descending order. It would go all the nodes with same number of height so it would hit the worst case which is O(n) where n is number of nodes.