Asuman Aydın

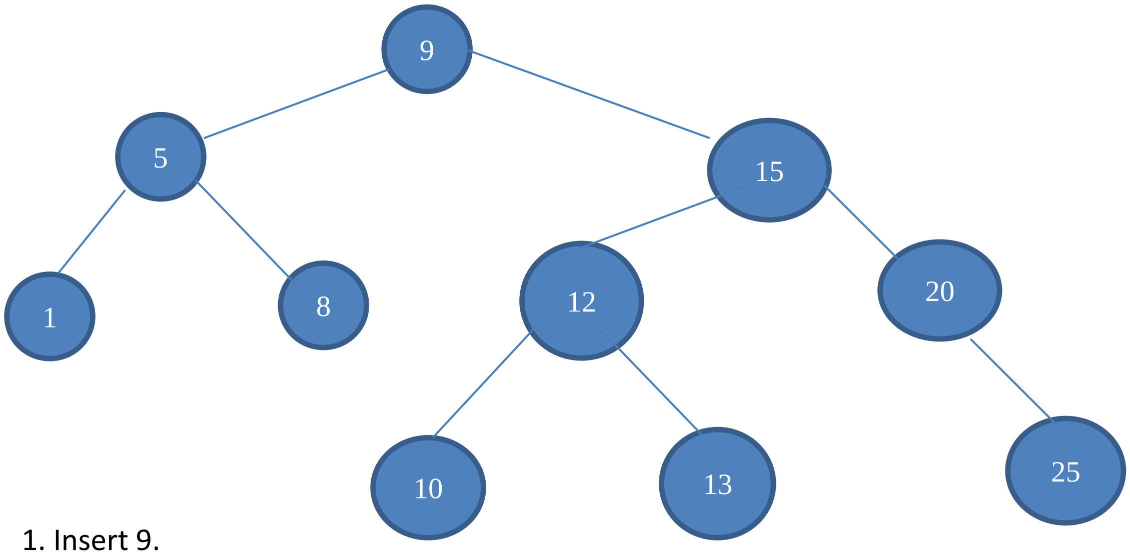
21502604

CS 202, Fall 2018

Homework #3 – Heaps and AVL Trees

Question 1:

a)



1. Insert 9.

2. Insert 12.

3. Check the parent and children so rotate right then left, insert 10.

4. Comparing the root and children, insert 5.

5. Comparing the root and children, it is smaller than all of them, insert 1.

6. Check balance for left, rotate right from 5, 9 is a child of 5 now.

7. 8 is smaller than 10 and bigger than 5 so check 9, it is smaller than 9 then insert it to the left child of 9.

8. As it is not balanced, rotate 9 to left and 10 to right.

9. 20 is larger than all so insert it to the right children of 12.

10. To keep the balance, rotate 12 to the left and make the 10 its left child.

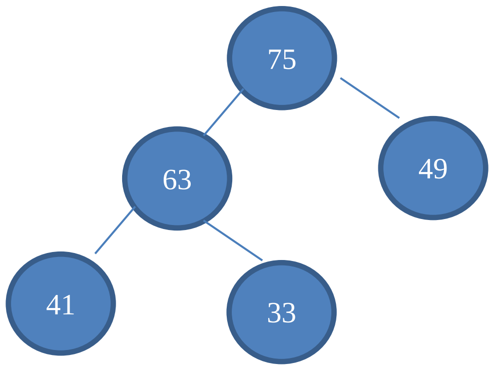
11. Insert 15 to 20's left child.

12. Insert 13 as 15's left child but to keep the balance rotate 15 to the right.

13. Insert 25 as a right child of 20 but to keep the balance rotate 15 to the left.

14. Insertion is done.

b)

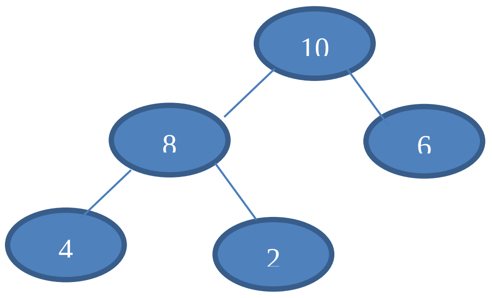


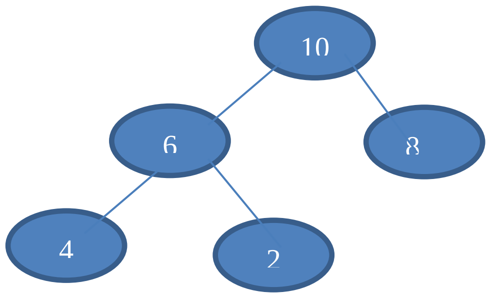
Heap Sort:

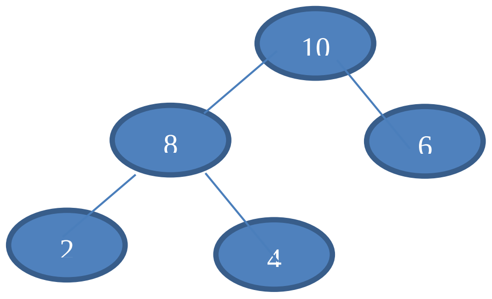
- ➔ 75 63 49 41 33
- ➔ Rebuild →33 63 49 41 75
- ➔ Swap → 63 33 49 41 | 75
- ➔ Rebuild → 41 33 49 | 63 75
- ➔ Swap → 49 33 41| 63 75
- ➔ Rebuild → 33 41| 49 63 75
- ➔ Swap → 41 33| 49 63 75
- ➔ Rebuild → 33| 41 49 63 75
- ➔ One element remained

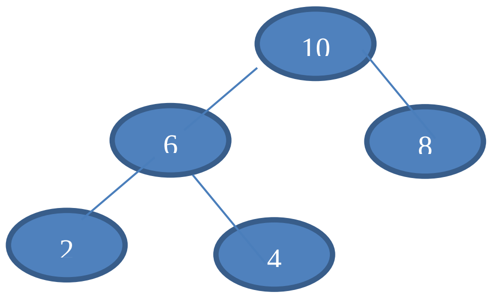
➔ Final: 33 41 49 63 75

c)

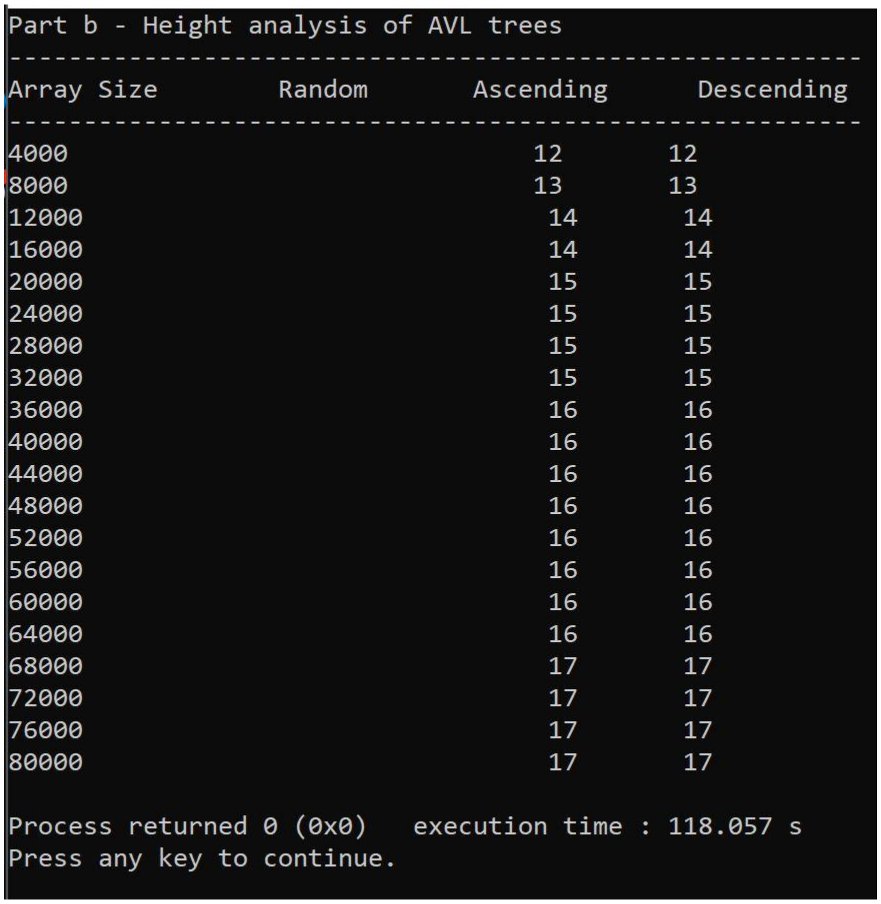








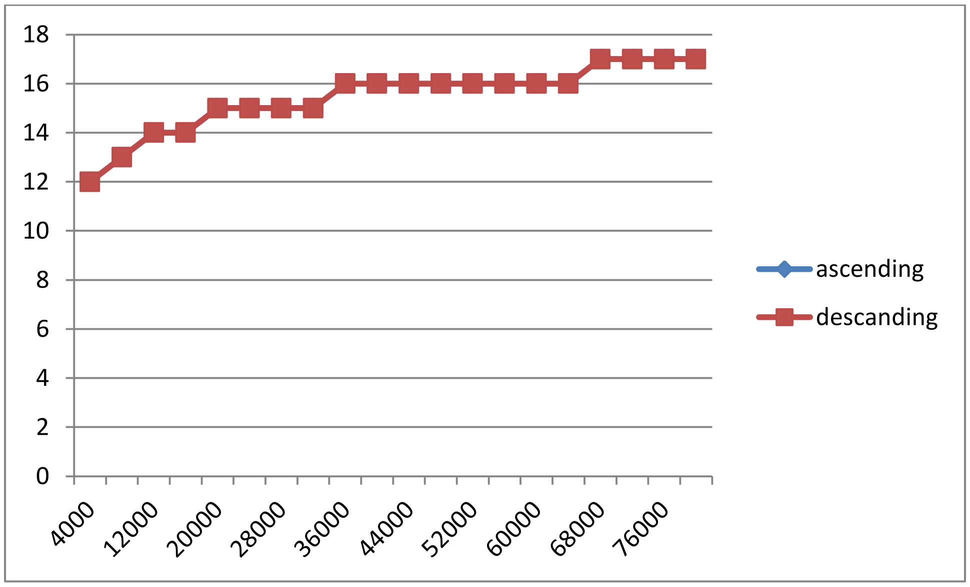
Question 2:



```
Part b - Height analysis of AVL trees
----------------------------------------------------------
Array Size          Random        Ascending       Descending
----------------------------------------------------------
4000                                  12        12
8000                                  13        13
12000                                   14        14
16000                                   14        14
20000                                   15        15
24000                                   15        15
28000                                   15        15
32000                                   15        15
36000                                   16        16
40000                                   16        16
44000                                   16        16
48000                                   16        16
52000                                   16        16
56000                                   16        16
60000                                   16        16
64000                                   16        16
68000                                   17        17
72000                                   17        17
76000                                   17        17
80000                                   17        17

Process returned 0 (0x0)   execution time : 118.057 s
Press any key to continue.
```

Question 3:



I could not have the results of random numbers inserted array. After inserting 600 numbers, it basically stops inserting and I could not solve the problem about this.

However, I have the results of ascending order and descending order. They have the same result because ascending insertion is growing from the right side and it basically always does a left rotation. For the descending ordered insertion, it does the reverse of ascending and it always does a right rotation as it is inserting smaller numbers every time. So, in the end, they have the same height.

For the worst case scenario, I think, it would be the operating every rotation options in every time we insert a item. It may not be possible but if it has to trace every node and if there are empty nodes between leafs, it would be hard to insert and delete.