ESO207A : Data Structures and Algorithms
Assignment 3 Solutions

Team Name : Invariantly_Yours

Yatharth Goswami: 191178
Sarthak Rout : 190772
Devanshu Singla : 190274

November 5, 2020

I Problem 1

1.1 Part (a)

Programming related

1.2 Part (b)

Programming related

1.3 Part (c)

Programming related

1.4 Part (d)

The five values for height that were observed are:

13	12	11	14	15

Average: **13.0**
On the other hand, if we insert items numbered 1 to 100 into an empty and ordinary BST, we will get a right-leaning tree of height **100**.
We also ran the program 10000 times and observed the following results.

Ranges of height	Count
1-5	0
6-10	116
11-15	8958
16-20	923
21-25	3
26-100	0

We can see that assigning priorities to the nodes allow us to uniquely specify the BST that will be formed. Also, if we choose the priorities in a randomized way, there is high probability that we will end up getting trees which are balanced most of the times. We observe that by implementing a "Treap" we have a nearly balanced binary tree and have a height of the order of $log(n) = log(100) \approx 7$.
So, we have height of the treap of the order of $log(n)$, while for an empty and ordinary BST the height is in the order of n.

1.5 Part (e)

The five values for height that were observed are:

6	7	7	11	8

Average : **7.8**
Also running similar statistical tests on this array led to the following results.

Ranges of height	Count
1-5	0
6-10	9500
11-15	500
16-20	0
21-25	0

On the other hand, if we insert the numbers in an ordinary bst, we find that the height of the BST would be 5 and the BST formed turns out to be nearly balanced.
We still observe that the average height obtained by using randomized priorities is still much better than the worst case height *i.e.* 24 and is near to the best possible height *i.e.* $\lfloor log(24) \rfloor$. We also observe that for small values of total number of nodes, it is not necessary that assigning the priorities randomly will give us the best(min) possible height, but as we increase the number of nodes, we will find that the height of the tree formed using this will approach the best(min) possible height.