Name – Moin Khan, Student ID – 22101287, Section - 001, CS202 – HW1

Height vs. Number of Nodes

Unsorted Random Inputs:

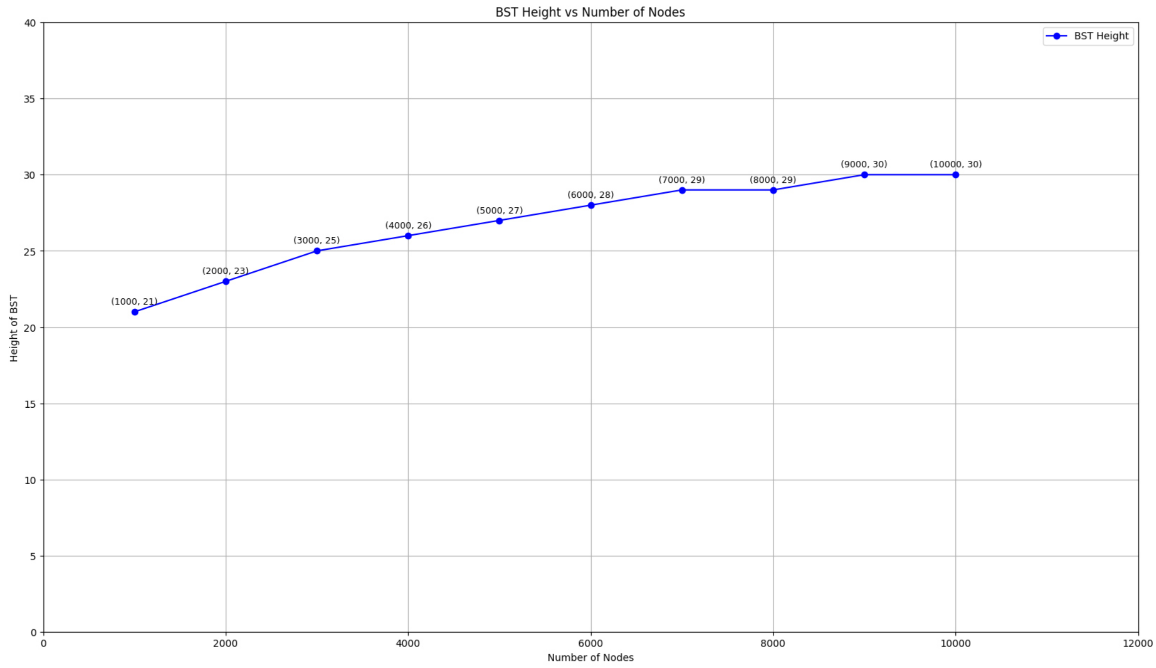


Figure 1: Height vs. Number of Nodes for every Thousand Insertion

Time vs. Number of Nodes, Unsorted Random Inputs:

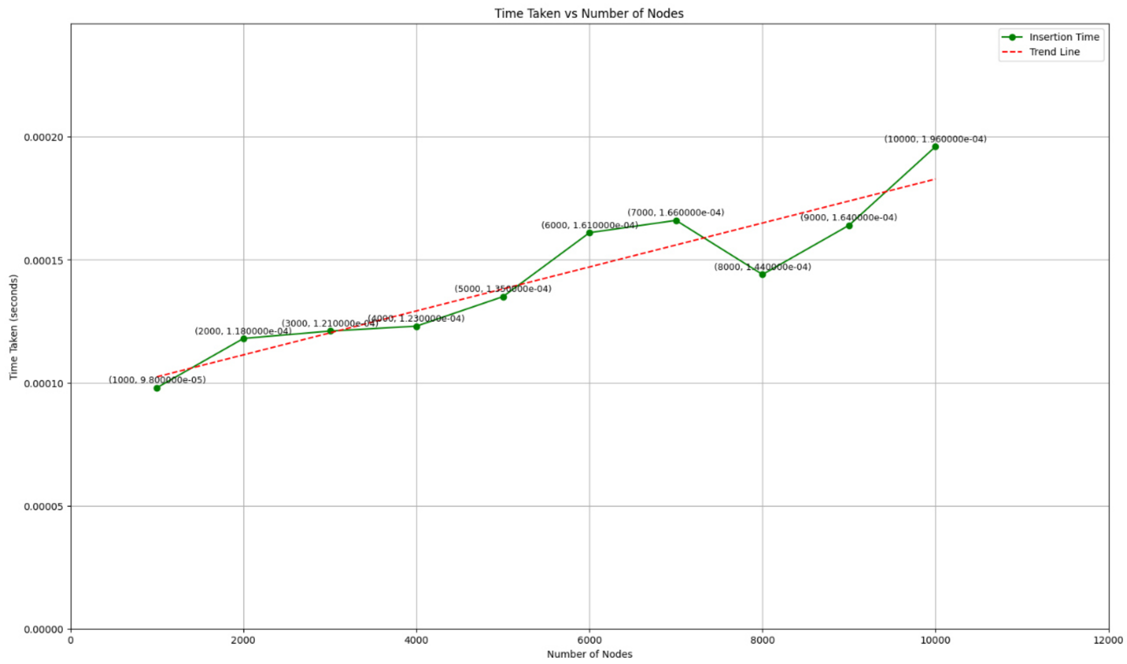


Figure 2: Time vs. Number of Nodes every Thousand Insertions

Question 3:

a: The height in the beginning increases rapidly when fewer nodes are present within the BST as the number of inserted nodes increases the growth of the height slows down which is expected. This graph resembles a logarithmic growth in height relative to the number of nodes, though its not perfectly logarithmic. This shows that our BST is reasonably balanced which is expected of randomly generated numbers.

b: The time vs nodes graph is fluctuating in the beginning but it settles down as the number of nodes increases, time is increasing as the number of nodes increases which is expected behaviour. If the data was sorted the behaviour would be of O(n) as the BST would degenerate into a linked list.

C: The suggested method of insertion for binary search tree would be to sort the data and divide it into 2 at the median and keep doing that for both sets and the insert the data around the median as the root. It would give the most balanced tree compared to random where its generally mostly balanced but can still go to O(n) time complexity in worst case scenario. Whereas in median method it would be O(log n).

H.W.1 | CS-202 | Name - Moin Khan

Id - 22101287

Question-1

(a) Second one represents correct pre-order traversal

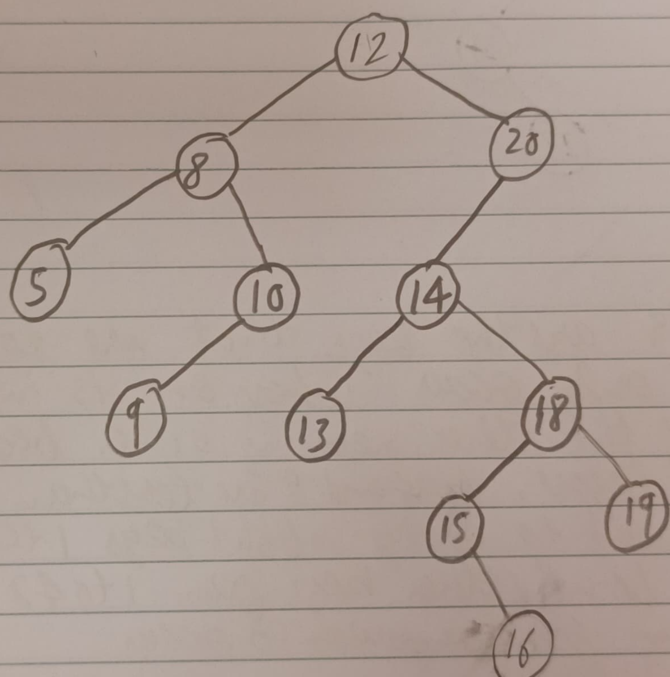
; < 12, 8, 5, 10, 9, 20, 14, 13, 18, 15, 16, 19 >

Post order

< 5, 9, 10, 8, 13, 16, 15, 19, 18, 14, 20, 12 >

In order

< 5, 8, 9, 10, 12, 13, 14, 15, 16, 18, 19, 20



(b) Given that a < b < c < d, we can only have b as the root. There will only exist possible ways of insertion starting with b as the root,

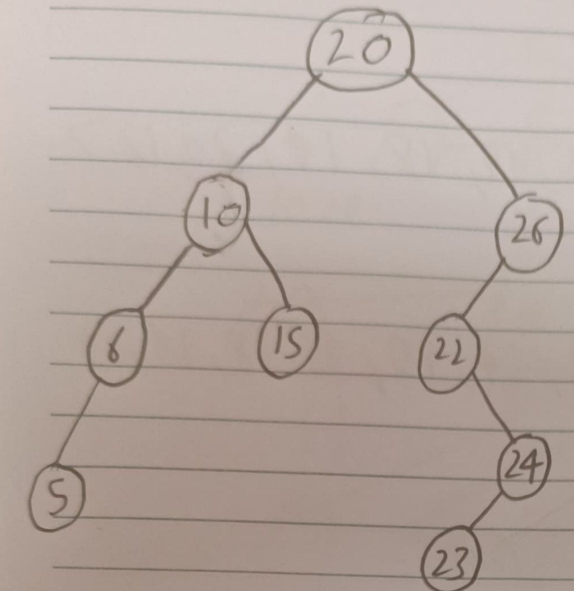
< b → a → c → d > or < b → c → a → d >

or

< b → d → a → c >

(c) pre-order: <20, 10, 6, 5, 15, 26, 22, 24, 23>

This gives the 7th key => $\boxed{22}$



(d) 4 and 9 are the keys that are to be compared, this can only occur when one is in the subtree of the other, meaning either becomes ancestor of the one before. As 4 and 9 are less than 43 we only need to worry about keys 1 through 42. There must be no other keys from 1 to 42 inserted between them for comparison to occur.
Three possibilities.

1) 4 is inserted before 9 and before any key that lies between 4 and 9.

2) 9 is inserted before 4 and before any other key that lies between 9 and 4.

3) Other 4 get inserted {5, 6, 7, 8}, which means we have 2 positive possibilities out of 6 possible insertions.

$$P(\text{comparison } 4 \text{ and } 9) = \frac{2}{6} = \boxed{\frac{1}{3}} \Rightarrow \boxed{33.33\% \text{ Chance}}$$

FABER-CASTELL