ESO207A

Programming Assignment 2

DEEPANSHU BANSAL(150219)
MUKUL CHATURVEDI(150430)

Comparing nearly balanced BST & perfectly balanced BST

For **n = 10,000** and **BALANCE_RATIO(BR)=0.75** we have ,

T(Insert(nearly balanced BST)) = 0.002223s

T(Insert(perfectly balanced BST)) = 0.002307s

Since worst case time complexity for insert is **O(h)** where h is height of tree.

Generally, for nearly balanced for n nodes and balance ratio x ignoring balancing we have

$$\begin{aligned} h(n) &\le 1 + h(xn) \\ &\le 1 + 1 + h(x^2 n) \\ &\dots\dots\dots\dots \\ &\le \log_{1/x} n + h(1) \end{aligned}$$

Thus we have h(n) = $\mathbf{O(\log_{1/x} n)}$

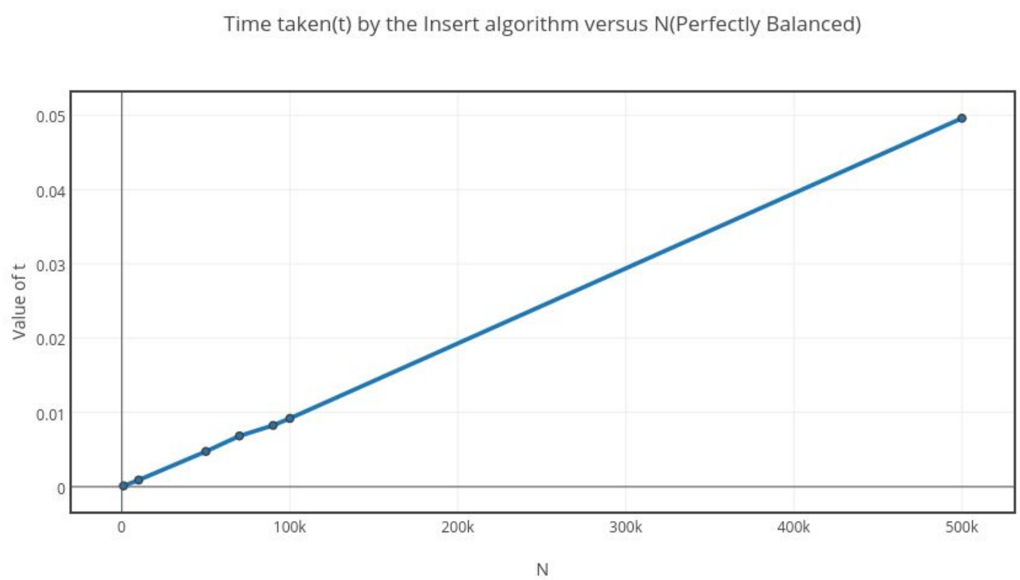
Thus time complexity for insert is

$$T(\text{ Insert(BR=x) }) = \mathbf{O(\log_{1/x} n)}$$

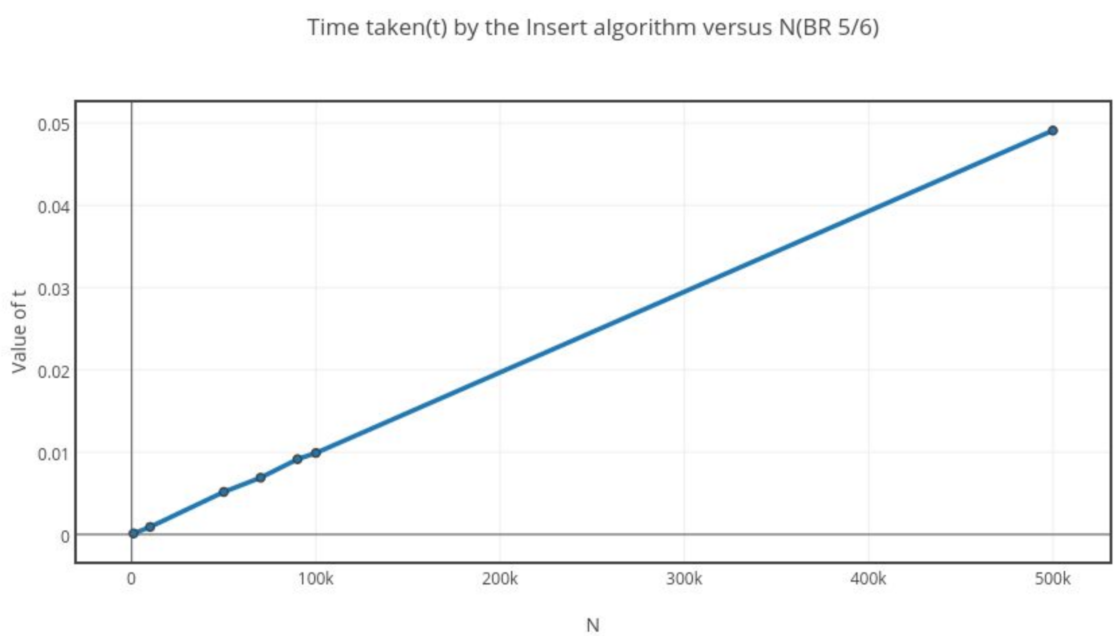
Hence time taken to insert in **perfectly balanced is more than nearly balanced** as we have to balance more number of times in balanced BST than in nearly BST hence more time required .

Plots

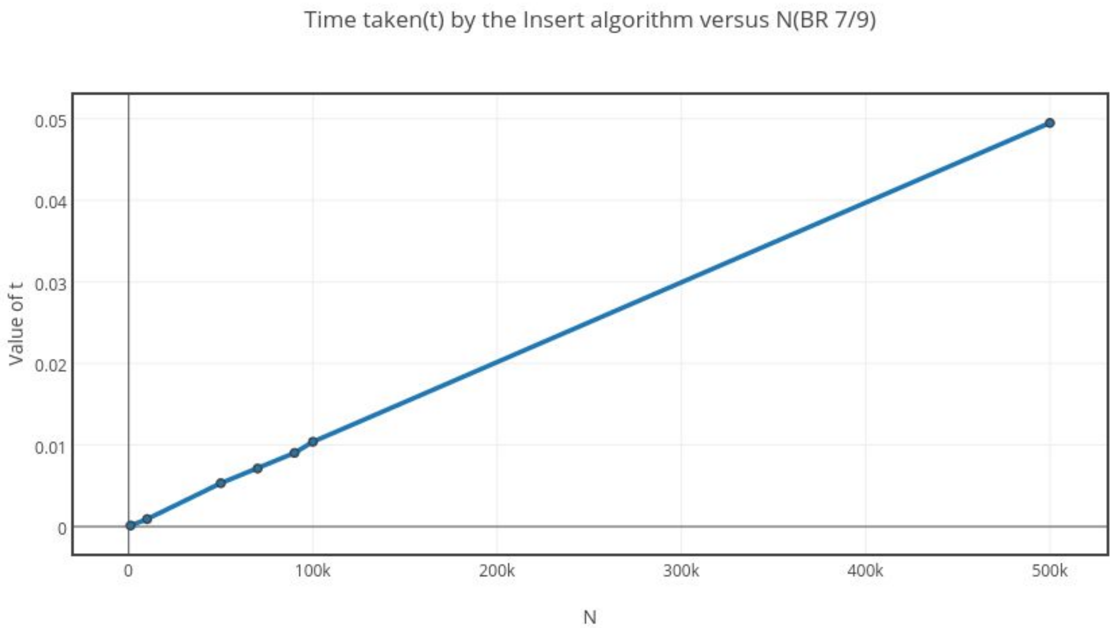
For **Perfectly Balanced case** we have T(Insert) vs N a **linear curve**



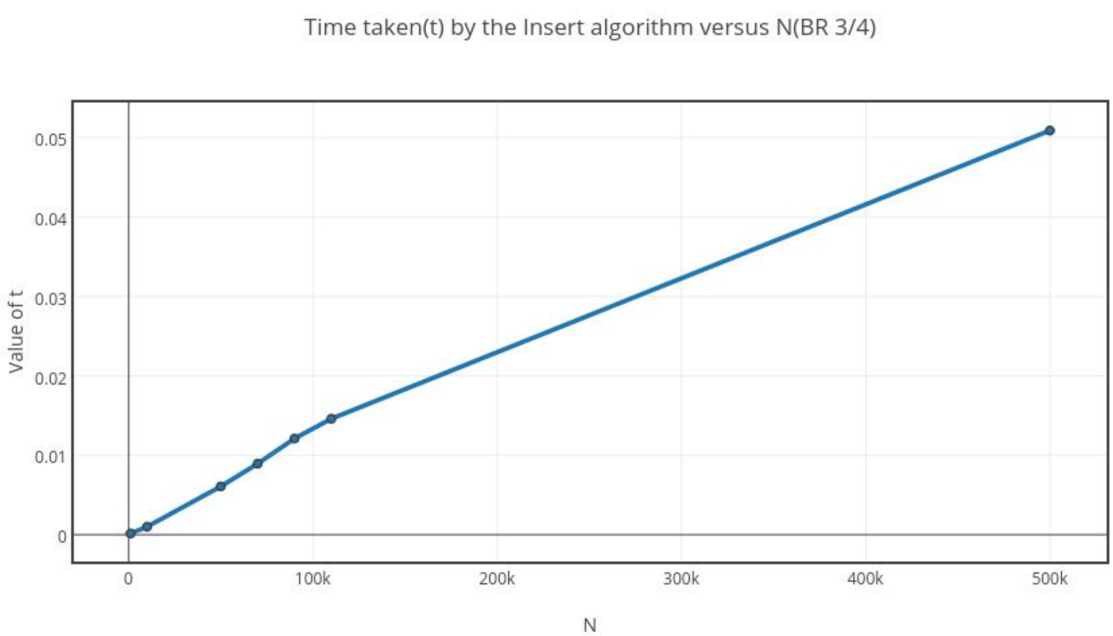
For **BALANCE_RATIO=5/6** we have we have T(Insert) vs N a **linear curve**



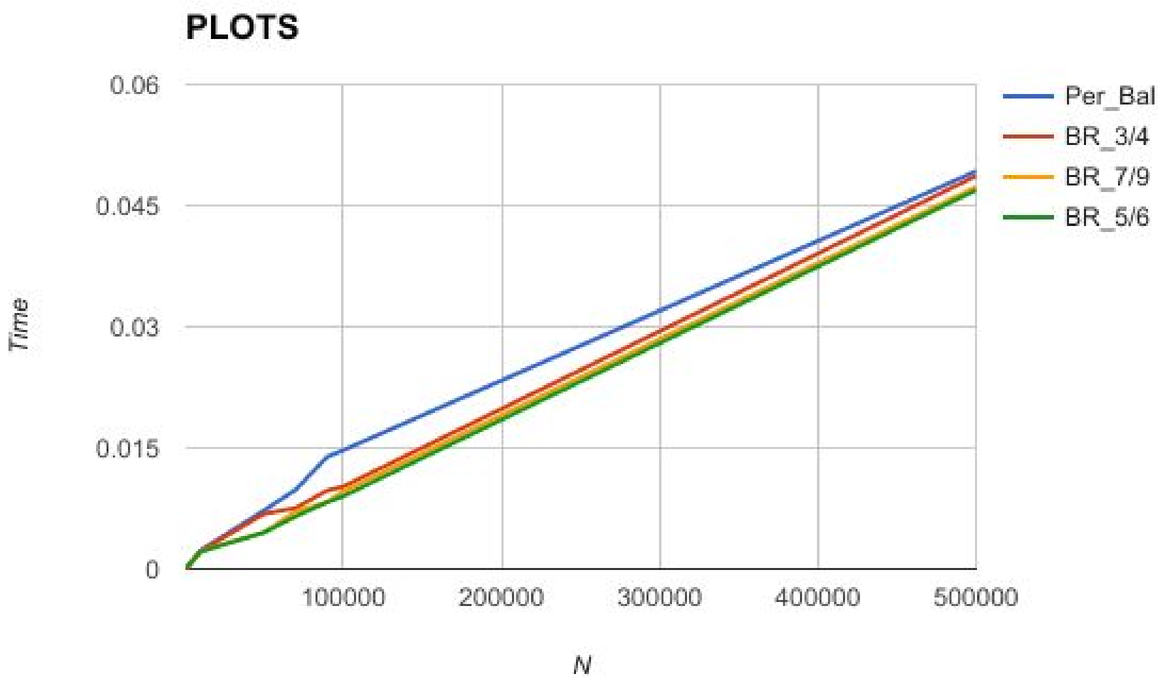
For **BALANCE_RATIO=7/9** we have we have T(Insert) vs N a **linear curve**



For **BALANCE_RATIO=3/4** we have we have T(Insert) vs N a **linear curve**



Collectively we have



Now in data form we have

Time is in seconds

BR	1000	10000	50000	70000	90000	100000	500000
1/2	0.000137	0.002307	0.007247	0.009808	0.0139	0.0147	0.049256
3/4	0.000119	0.002223	0.006841	0.007543	0.009724	0.0102	0.048685
7/9	0.000110	0.002210	0.004535	0.007114	0.008294	0.0096	0.047312
5/6	0.000108	0.002168	0.004514	0.006514	0.008274	0.0090	0.046922

ARGUMENTS

As n increases T(insert) also increases as T(Insert(BR=x)) = **O($\log_{1/x}$n)** derived previously.

This explains the linear nature of all graphs.

Now for relative comparison we have

As tree becomes more and more imbalanced ie BR increases it becomes easier to insert a value into the tree so total number of rearrangements of subtree at some node becomes less and less as BR increases which is self evident from the data and graph .