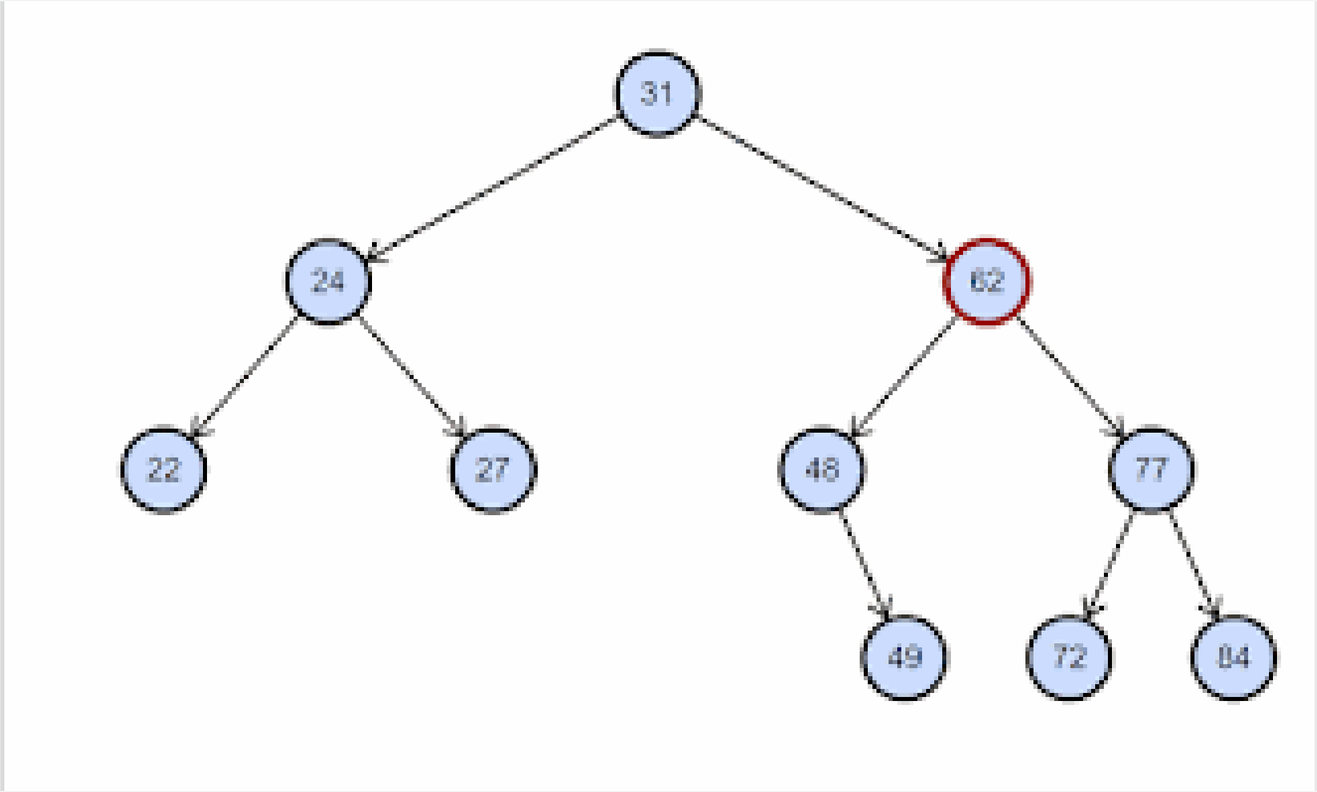
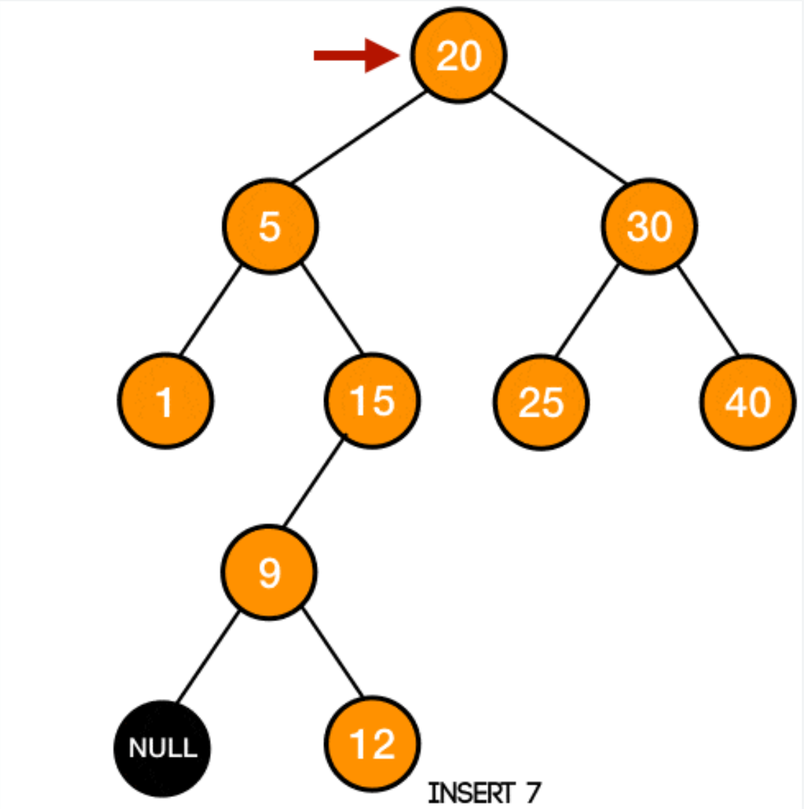
ANALYSIS OF DIFFERENT BINARY SEARCH TREES

PROJECT PROBLEM

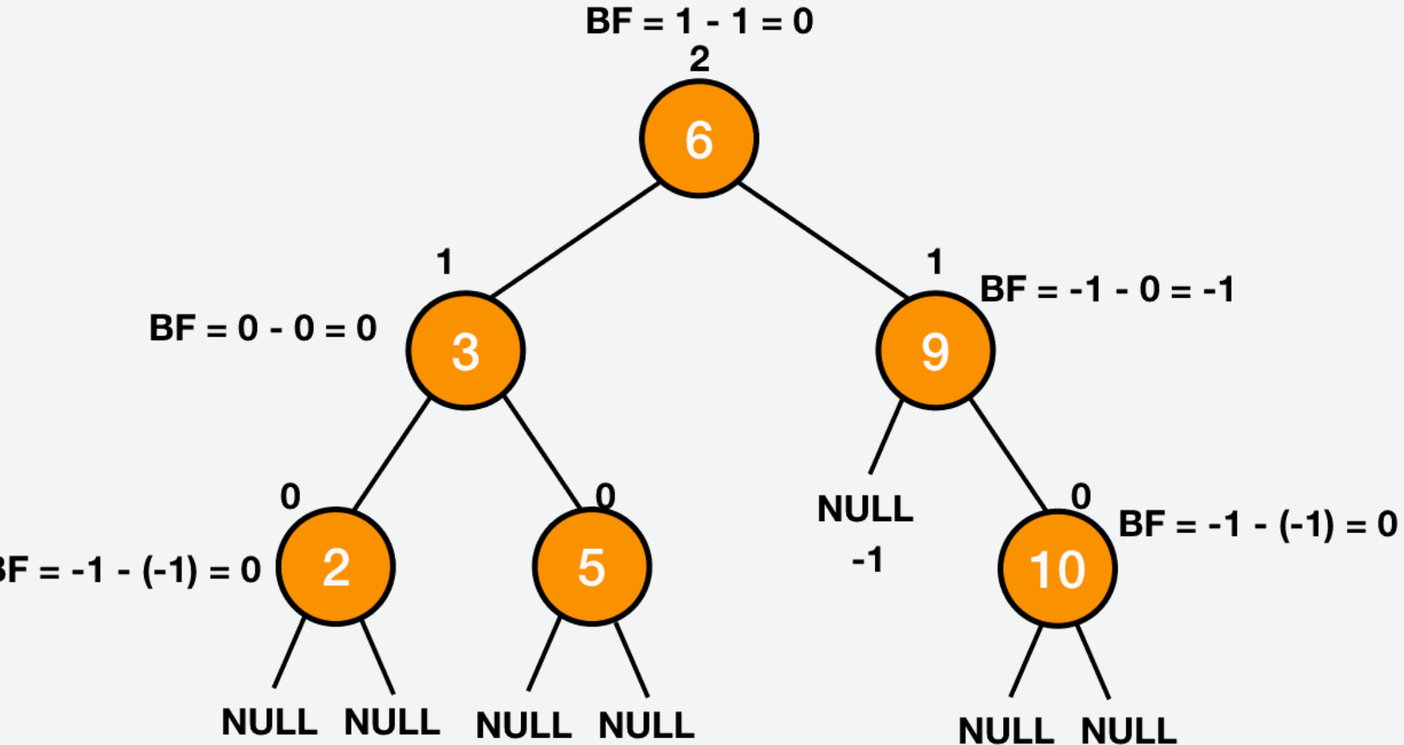
AVL , Red-Black and Scapegoat Trees are all self balancing binary search trees. Each of these trees use a different approach to balance themselves. We compared and assessed the three structures under various scenarios such as node inserting, deletion and searching





AVL TREE

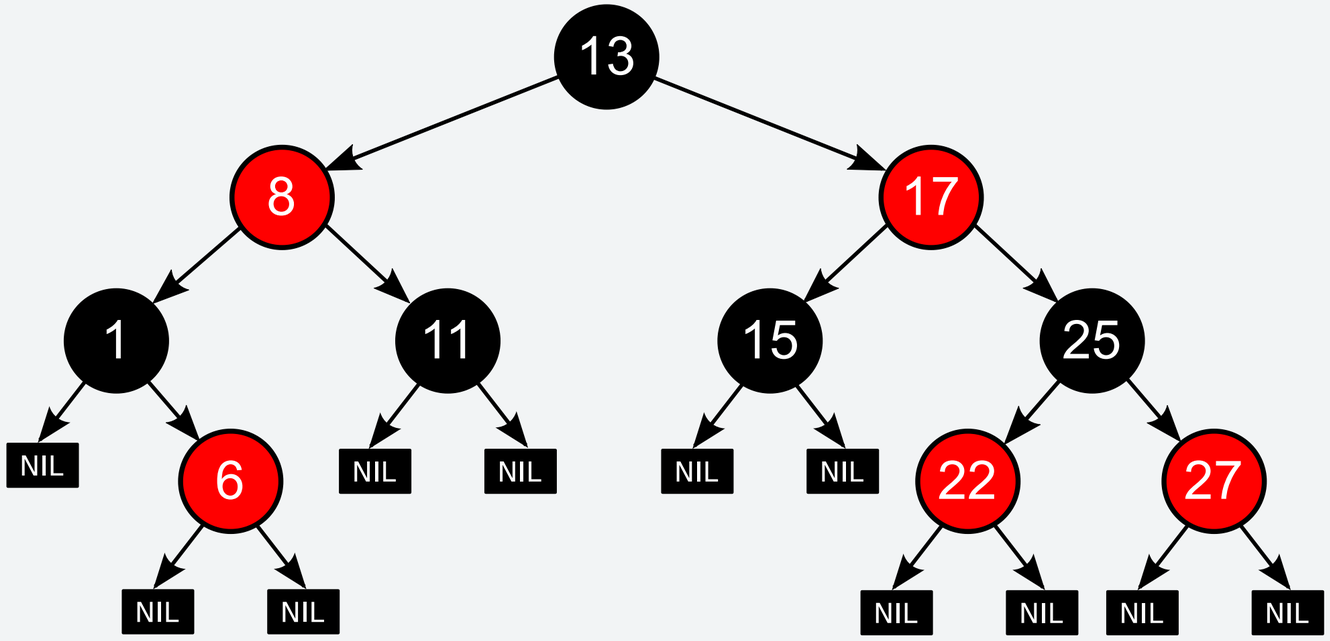
- self balancing binary search tree
- the height difference between the left and the right subtree of a node (balance factor) should be at most 1
- if under any circumstances balance factor differs by more than 1, single or multiple rotationsare applied to restore this property.



	Average	Worst
Insertion	O(logn)	O(logn)
Deletion	O(logn)	O(logn)
Searching	O(logn)	O(logn)
Traversal	O(n)	O(n)

RED-BLACK TREE

- a self balancing binary search tree
- each node has one extra bit which represents it's color (**red** or **black**)
- the root is always **black**
- the children of the **red** nodes are always **black**
- Insertions and deletions may require tree rotations and recoloring to maintain these properties.

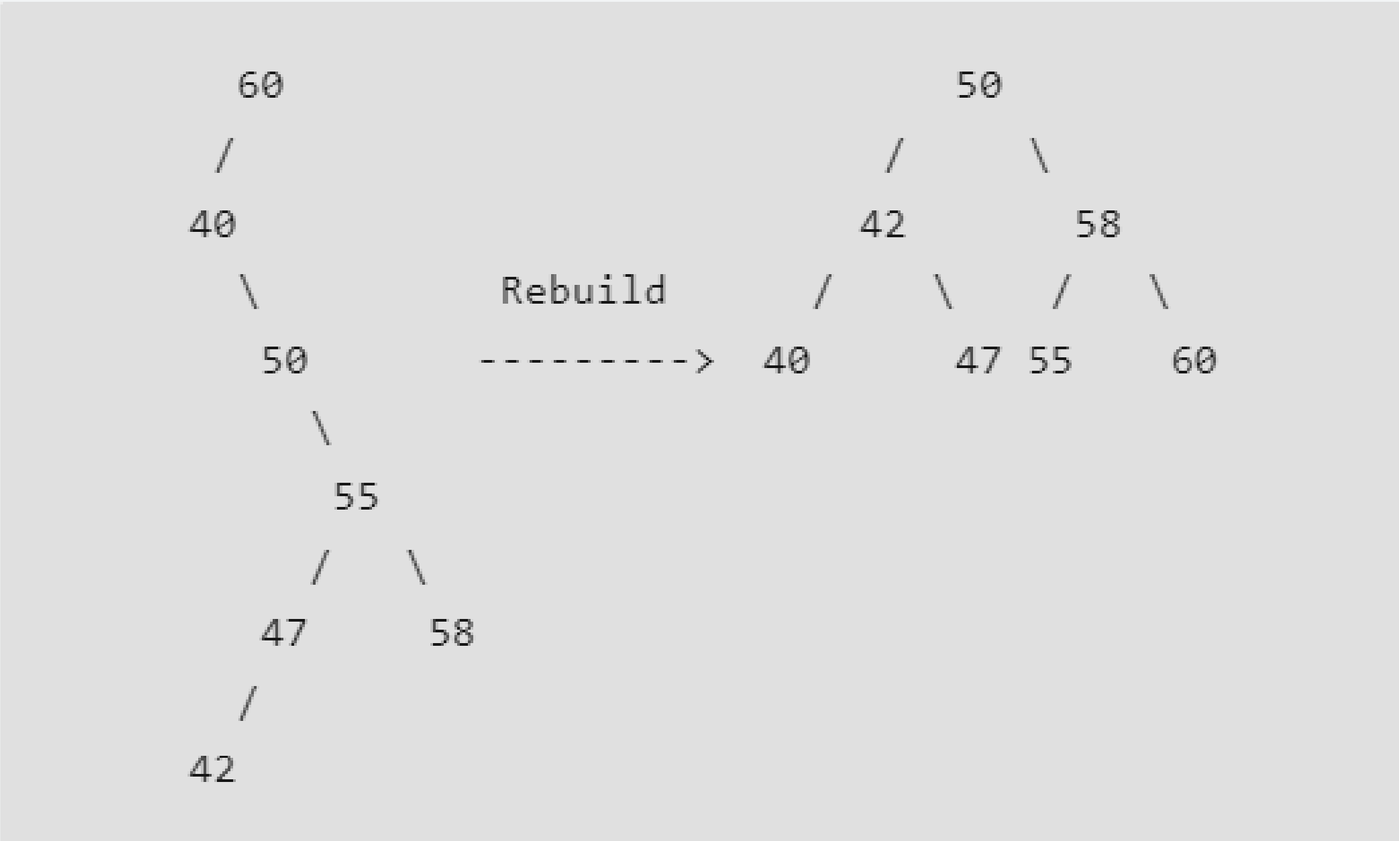


Operation	Average Case
Search	O(log n)
Insertion	O(log n)
Deletion	o(log n)

SCAPEGOAT TREE

- self balancing binary search tree
- balances itself through occasionally rebuilding its unbalanced subtrees
- when a subtree becomes unbalanced beyond a certain threshold α, the subtree is rebuilt

Search	$O(\log_2(n))$
Traversal	$^*O(n)$
Insert	$^*O(\log_2(n))$
Delete	$^*O(\log_2(n))$



APPROACH

We generated a dataset comprising 20,000 strings as our input and constructed three different types of trees: AVL tree, red-black tree, and scapegoat tree. We then executed insertion, deletion, and searching operations on each tree and meticulously measured their respective time complexities.

RESULT

Comparing the insertion, deletion and searching for our dataset, we have the following statistics

- For insertion, AVL took 41,658.7 , Red-Black took 28,553.4 and scapegoat took 27,059.7 microseconds so scapegoat tree takes the lead for this one
- For deletion, AVL took 15.8, Red-Black took 5.833 and scapegoat took 10 microseconds so Red-Black tree tree took the least amount of time deleting a node
- For searching, AVL took 7.7, Red-Black took 4.777 and scapegoat took 6.9 microseconds so Red-Black tree takes the least amount of time searching for a node

THANK YOU FOR YOUR TIME

Group Members:
Adina Adnan Mansoor
Mahrukh Yusuf
Sameer Kamani
Shifa Shah