Title: Study Splay tree and compare wrt Red Black tree

Team members: K Yaswanth (2018202011), SVL Sarat chandra (2018202013)

Deliverables:

C++ code for splay tree and Red black tree, graphs for various test data for insertions, deletion and search operations.

Project delivery plan:

[X] Understand project requirements.

[X] Study of Splay tree and Red black tree.

[ ] Building code for splay and red black tree.

[ ] Performing tests and plotting graphs.

[ ] Project report.

Technologies:

C++ (for building project) and python (for visualization).

Online resources:

1. https://en.wikipedia.org/wiki/Splay_tree
2. https://en.wikipedia.org/wiki/Red%E2%80%93black_tree
3. http://www.cs.cmu.edu/~sleator/papers/self-adjusting.pdf

Repository where work is being committed:

https://github.com/yaswanthkoravi/Apsproject.git

Plan for testing:

Testing phase is divided in to three components.

1. Insertion of nodes:

   For this test, we randomly generate certain numbers and measure time taken for inserting nodes on both splay and red-black tree and plot graph accordingly.

2. Deletion of nodes:

   Similar to insertion test, we measure time taken for deletion of nodes from both tree and plot graph accordingly.

3. Search for nodes:

This test is further divided in to two more components

3.1 Search for similar set of nodes:

In this test, after inserting randomly generated values on both trees, a subset of nodes (say 100) selected from inserted nodes and search operation is performed repeatedly (say 1000 times) on both splay and red black trees. We then measure time taken to perform each search operation and plot graph accordingly.

3.2 Search for random nodes:

In this test, we select a random node and search operation is performed on both trees. We then measure time taken to perform each search operation and plot graph accordingly.

End user documentation:

After executing the code, the end user is asked to choose one of the following operations to be performed on both red-black and splay tree

1. Insertion of a node
2. Deletion of a node
3. Search for a node