

6. Splay tree implementation and its comparison with AVL tree

Team members:

Vatsal Soni	2018201005
Darshan Kansagara	2018201033

Deliverables: Implantation of insert, search and delete operation in splay tree and its comparison with AVL tree. Graphical in-depth analysis of splay tree and AVL tree.

Project delivery plan:

- Phase 1. Understanding of all operation in splay tree.
- Phase 2. Implementation of insert, search and delete operations in splay tree.
- Phase 3. Find different between Splay tree and AVL tree.
- Phase 4. Graphical in-depth analysis of splay tree and AVL tree.
- Phase 5. Find some real world application of Splay tree and AVL tree.

Technologies to be used:

Language: C++

Online resources:

1. https://en.wikipedia.org/wiki/Splay_tree
2. <https://www.cs.usfca.edu/~galles/visualization/SplayTree.html>
3. <https://www.youtube.com/watch?v=IBY4NtxmGg8>
4. <https://plot.ly/create/#/>
5. <https://web.stanford.edu/class/archive/cs/cs166/cs166.1146/lectures/08/Small08.pdf>

Repository link:

<https://github.com/darshank15/splaytree>

Plan for testing:

1. After every operation make sure that inorder traversal of a tree must be sorted.
2. Comparing output of insertion and deletion with visualization.
3. Using random generator file we will generate different sets of input and will perform insertion and deletion on splay tree and AVL tree (try to cover all corner cases).
4. We will test two type of search operation i.e. Random search and frequent element search.

- a. For random search we will generate random numbers and insert it into the tree and store it in array/vector. We will randomly choose some elements from array/vector and perform search operation.
- b. For frequent search we will choose only 10 elements randomly and then we will perform search operation many time (thousand time) on splay tree to search element from above sets (from 10 elements).

End user documentation:

1. Giving brief introduction about splay tree, AVL tree and its operations.
2. Giving abstract definition of insert, delete and search operation.
3. Providing graphical representation of all operations in terms of time complexity of splay tree and AVL tree.
4. Try to provide areas where splay tree and AVL tree can be used in real word to achieve better efficiency.