Persistent Segment Tree

Kshitij Paliwal 2018201063

Sivanagi Singh 2018201001

- **Abstract :** Persistence is a concept in data structures, where we reuse the original state of data structure whenever some update is required. This results in a data structure where we can see the state of data structure before and after the update with minimal use of memory. Segment trees are interval trees over an array where we can store informations about segments. It allows us to update the information regarding some segment of the array in O(log(n)) time complexity.

- **Deliverables:** Create a **persistent segment tree** where we should be able to get the info of any segment after any update performed.

- **Project Delivery Plan:**
  1. Understanding & Learning Segment Tree
  2. Implementation of Segment Tree
  3. Understanding & Learning Persistent Data Structure
  4. Understanding & Learning Persistent Segment Tree
  5. Implementation of Persistent Segment Tree
  6. Implementation of Generalised Persistent Segment Tree
  7. Testing of Implemented of Persistent Segment Against Standard Implementation of Persistent Segment Tree
  8. Analysis of Persistent Segment Tree against Segment Tree with respect to time and space complexity

- **Technologies to be used:** C++ & Plottly API

- **Online Resources:**
  - Wikipedia: https://en.wikipedia.org/wiki/Persistent_data_structure https://en.wikipedia.org/wiki/Segment_tree

- Youtube:https://www.youtube.com/watch?v=TH9n_HVkjQM

- **Repository:** https://github.com/kshitij02/Persistent_Segment_Tree.git (Private Repository)
- **Plan for testing :**
  - Run multiple test cases on Presistent Segment Tree to check whether it gives result according to the standard implementation of Presistent Segmemt Tree.

- **End user documentation :**
  - Provide the size array
  - Elements of Array
  - Type of Segment Tree :
    - min
    - max
    - sum
  - Type of Operation to Prefrom :
    - Query
    - Traversal
    - Update
    - Find kth Min/Max