Sample text file will look like:

**10**

**2 5 -3 6 1 2 9 -4 7 8**

Output the SegmentTree in DOT format to a filet:

**digraph G {**

**[label="L"]**

**[label="2"]**

**[label="R"]**

**[label="5"]**

PseudoCode:

Create header and cpp files for this program

Create a segment tree and node class

Create insert and search functions for tree

Will create a dot file for the end

Use input text file to produce results

Node.h:

// Constructor for a node with a given start, end, and sum.

// Node pointer for left and right

// Destructor for a node, deletes the left and right subtrees.

// Private variables for int, start, and sum

Node.cpp:

// Implementation of node class

Segment.h:

// Constructor takes an input array and builds a segment tree for it.

// Initialize a insert and search method

// Search for the sum of the values in the range [start, end] of the original array. // Insert a value at a specific index in the original array and update the segment tree. // Write the structure of the segment tree to a Graphviz file.

Segment.cpp:

// get the size of the input array

// create a new node as the root of the segment tree

// loop through each element in the input array

// create the tree using insert method

Main.cpp:

// command line argument parsing: we need a filename to pass to the SegmentTree class

// some error handling?? (maybe try to catch *invalid filename entered*)

Research Points:

How segment tree works and what is it

How insert works

How search works

How dot file works

1. Create github
2. Format code for respective .h and .cpp

2.5. Add potential remove function

1. Gather ideas for report/ follow guideline