EECS 214/395
Name: Surabhi Ravishankar

PROJECT 2 REPORT

EXECUTION:

$ make
$./Project2 input.txt

DESIGN & THOUGHTS:

The binary search tree insert , remove and the PR Quad Tree insert and remove were the most challenging parts of this assignment. I used the following algorithm to implement binary search tree insert. At each node I checked if the data to be inserted is greater than the left child or right child and then called a recursive insert function. The node gets inserted to the left if it's smaller than the current node you are comparing with. The algorithm that I used is given in the comments of bst.cpp file.

I implemented the Binary Search Tree remove function by checking 3 cases. First I check if the node has only a left child, then I checked if the node has only a right child. In both the cases I can just delete the node. In case of Nodes with 2 children, I push the node down, by exchanging it with the minimum and then delete it.

Next, I implemented the PR Quad Tree. I implemented classes in the following way, I have used Inheritance in order to implement the PR Quad Trees. This consists of a base class "qnode" which have sub classes for internal nodes and external leaf nodes, "qleafnode" and "qinternal" node. The leaf nodes and the external nodes store information like the spacial direction of the coordinate system. Stores the parent node and the number of children.

The PR Quad Tree's Insert function was implemented in the following way,
First I found the location of the coordinates location location in the space, If the coordinate is null I insert a node, keep the current node as root and insert the new node its new subtree. The function also handles duplicate entries.

The PR Quad Tree's remove function was implemented in the following way, First I find the location of a node, that contains the record for the cityName. If it is an external node then I just delete it. Next, the node's siblings are checked, and might need to be merged. Merging occurs when there is only one coordinate. The merge operation takes place until a level where there are 2 points in a subtree. I found this the toughest to implement.

Implementation and Algorithm details for the rest of the functions are given in detail in the program files.