

Solution

Binary Search Tree:

I store the city name, x coordinate and y coordinate in the binary tree. Every node has three pointers: left, right, and down. The left pointer stores the left child of that node as well as the right one. For those nodes, which have equal name, I store them by using the down pointer. The tree is sorted by the name of nodes. There are three main functions in the Binary Search Tree. They are insert, find and delete.

For find function, I return the reference of the target node. In the delete function, if there are two children in the target node, then I replace the target node with the min node in the right subtree.

PR-Quad Tree:

There are three types of its nodes. The QTreeNode is the abstract father node. The internal node and the leaf node are inherited from the father node. There are also three main functions. Besides, there is an assistant function called location to locate the direction (nw, ne, sw, se) of one node.

For the insert function, to begin with, if the root is null, then store the node in the root without divide into four parts. Otherwise, using location function to locate the direction of the target node in the internal node. Because of recursive, you can reach the target leaf node where you want to store the new node. If the node is empty, then just store the information. Otherwise, if the node is not null, and the information of two nodes are same, warning cannot insert the node. Otherwise, let the leaf node become an internal node and store the data.

For the delete function, to begin with, figure out the target node. If there is only one node after the deletion, then I turn the father node into a leaf node.

For the search function, I just search all nodes in the tree and output nodes in the target region.

Deal with input txt:

I use `getline()` method to get the content of one line, then extract the command split by space. And conduct functions according to the command getting from the input file.