GIT Department of Computer Engineering
CSE 222/505 - Spring 2022
Homework 5 Report

Student Name: Enes Abdülhalik
Student Number: 1901042803

Question 1.a:

Since a complete binary tree with height h is a perfect binary tree until the level h – 1, the total number of depths of the nodes is between two perfect binary trees with the heights h and h – 1 respectively. So, the total of depths of the nodes of a perfect binary tree with height h is:

$$D(h) = \sum_{n=1}^{h} n2^{n-1} = (h-1)2^h + 1$$

Then, the total of depths of the nodes of a complete binary tree is:

$$(h-2)2^{h-1} + 1 < D(h) \leq (h-1)2^h + 1$$

The exact value depends on the number of nodes in the final level, in which for a complete binary tree with n elements and height h, the total of depths of the nodes is:

$$D(h,n) = h\left(n - 2^{h-1} - 1\right) + \sum_{n=1}^{h-1} n2^{n-1} = h\left(n - 2^{h-1} + 1\right) + (h-2)2^{h-1} + 1$$

$$= h(n+1) - 2^h + 1$$

Question 1.b:

The average number of comparisons for a successful search operation in a binary search tree is the total of depths of the nodes of the complete binary search tree divided by the total number of nodes. So, for a complete binary search tree with n nodes and height h:

$$(total\ depth) \quad D(n,h) = h(n+1) - 2^h + 1$$

$$(average\ number\ of\ comparisons) \quad C(n,h) = \frac{D(n,h)}{n}$$

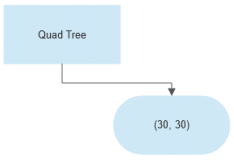
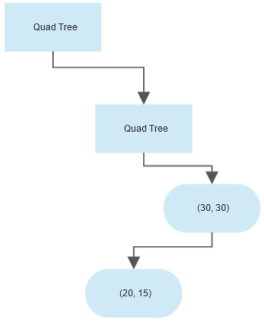
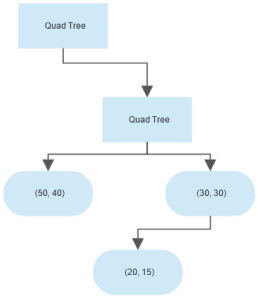
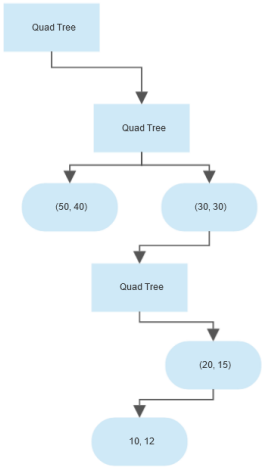
Question 1.c:

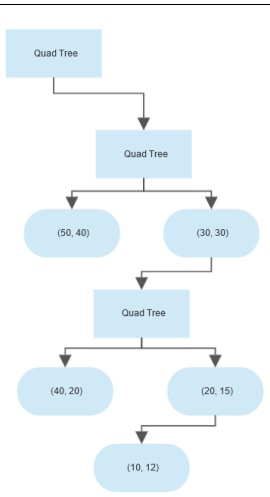
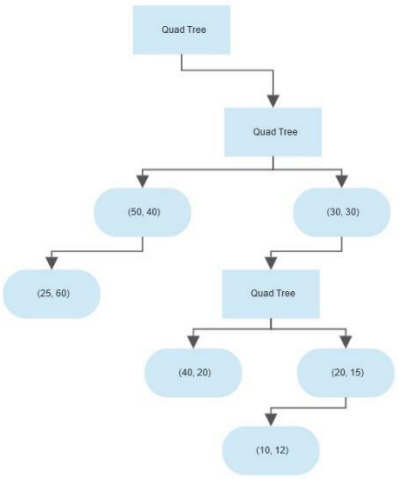
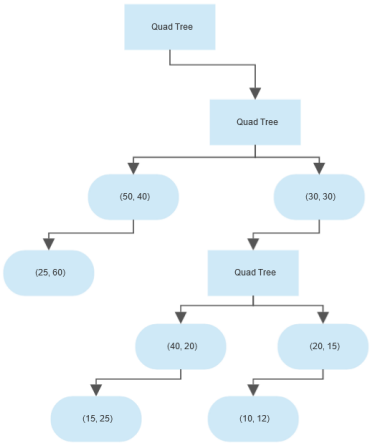
The total number of nodes cannot be an even number, because the nodes must be added 2 at a time for all the internal nodes to have 2 children, and the total starts from 1 (root), so the number of nodes has a restriction.

Experimentally, every full binary tree with n elements has $\frac{n+1}{2}$ leaves, and $\frac{n-1}{2}$ internal nodes.

Question 2:

Note: The left child is an element that exists in the same box but in another quarter. The right child is an element inside the tree.

Input	Steps	Result
1- (30, 30)	- Root is empty, insert directly in it.	
2- (20, 15)	- Traverse to the right side. - It's a point in the same quarter as the target. - Replace (30, 30) with a tree, then add both points to the new tree.	
3- (50, 40)	- Traverse to the right side. - It's a tree and it's not in the target's quarter. - The left side is empty, insert directly in it.	
4- (10, 12)	- Traverse to the right side. - It's a tree, and it's in the target's quarter. - Traverse to the right side. - It's the point (30, 30), and it's not in the target's quarter. - Traverse to the left side. - It's the point (20, 15), and it's in the target's quarter. - Replace (20, 15) with a tree, then add both points to the new tree.	

5- (40, 20)	- Traverse to the right side. - It's a tree, and it's in the target's quarter. - Traverse to the right side. - It's the point (30, 30), and it's not in the target's quarter. - Traverse to the left side. - It's a tree, and it's not in the target's quarter. - The left side is empty, insert directly in it.	
6- (25, 60)	- Traverse to the right side. - It's a tree, and it's not in the target's quarter. - Traverse to the left side. - It's the point (50, 40), and it's not in the target's quarter. - The left side is empty, insert directly in it.	
7- (15, 25)	- Traverse to the right side. - It's a tree, and it's in the target's quarter. - Traverse to the right side. - It's the point (30, 30), and it's not in the target's quarter. - Traverse to the left side. - It's a tree, and it's not in the target's quarter. - Traverse to the left side. - It's the point (40, 20), and it's not in the target's quarter. - The left side is empty, insert directly in it.	

Question 3:

Class Diagram: