GIT Department of Computer Engineering

CSE 222/505 - Spring 2022

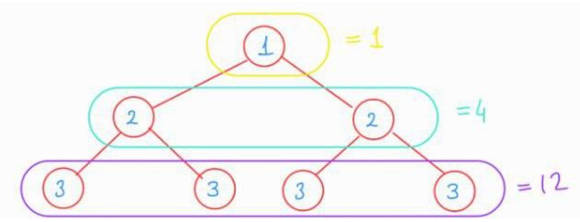
Homework 5 Report

Mustafa MERT

200104004006

Q1)

a)

if 

For every row, the common total depth formula is $h * 2^{h-1}$ while h is equal to the height of that row. According to this formula we want to find the maximum depth for height h in a complete tree the expression should be $\sum_{n=1}^{h} n * 2^{n-1}$.

If we want to find the minimum depth for height h in a complete tree the expression should be $(\sum_{n=1}^{h-1} n * 2^{n-1}) + h$.

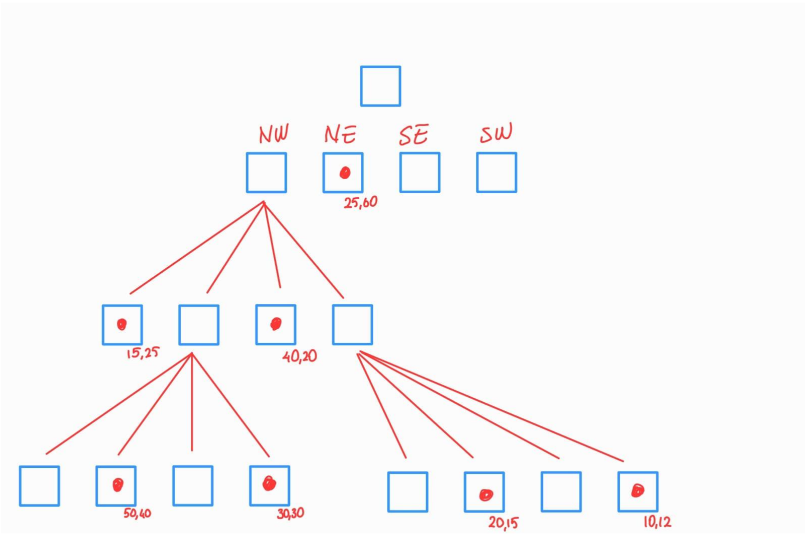
b) Searching an element in a binary search tree is efficient, why? Where the binary search tree is in the complete binary tree structure, the complete binary tree structure is the perfect binary tree through the before last level some extra leaf nodes at last level all toward the left. When you are searching an element you are comparing with the root then according to the result you are searching the half of the tree like in the binary search algorithm. Because of that time complexity will be logn.

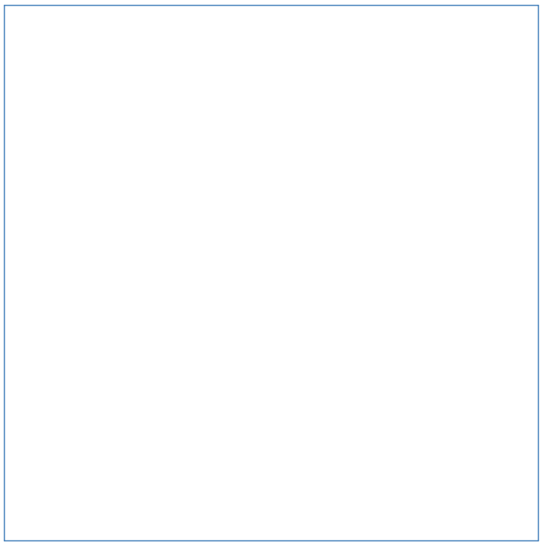
In a binary search tree if the searching item at level h we will do h comparisons. The probability for each item being the searched key = 1/n where n equals to the number of nodes in the tree. The result should be $\sum_{i=1}^{n} h_i \times \frac{1}{n}$ where h_i is the height of the node itself.

c) There is a restriction for full binary trees which is that all internal nodes must have 2 children. For **n** nodes in a full binary tree there are **(n-1)/2** internal nodes and **(n+1)/2** leaves.

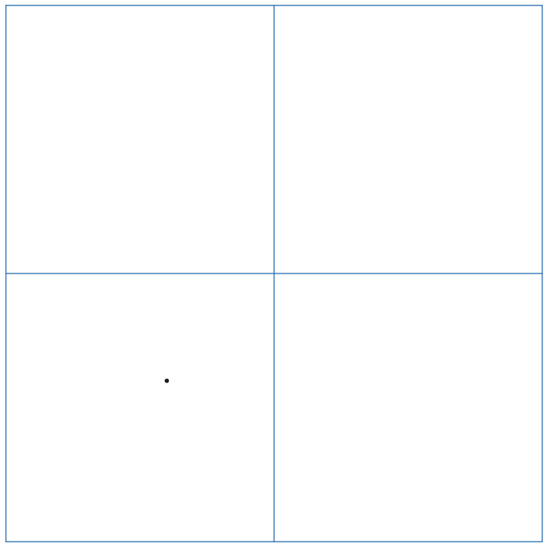
Q2) In quadtree there are northwest, northeast, southwest and southeast nodes for every node. When new data comes, according to the first data we put the suitable node, if this node is not empty we compare with that node and it goes like that until we find a free suitable node.

inserting given data

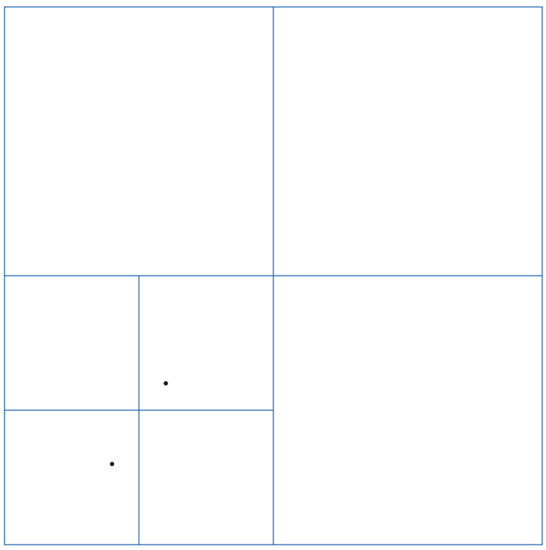




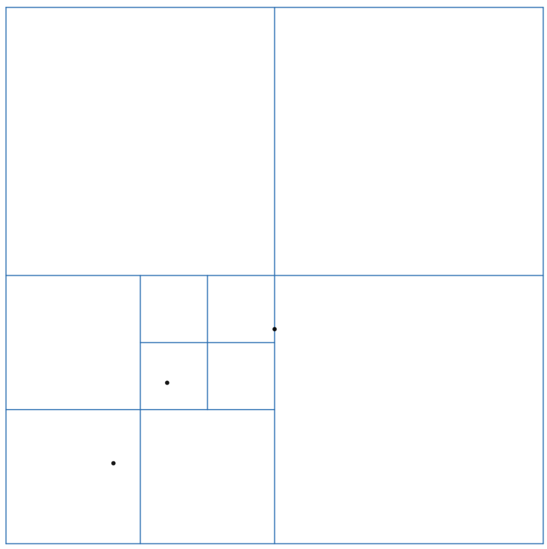
(30,30)



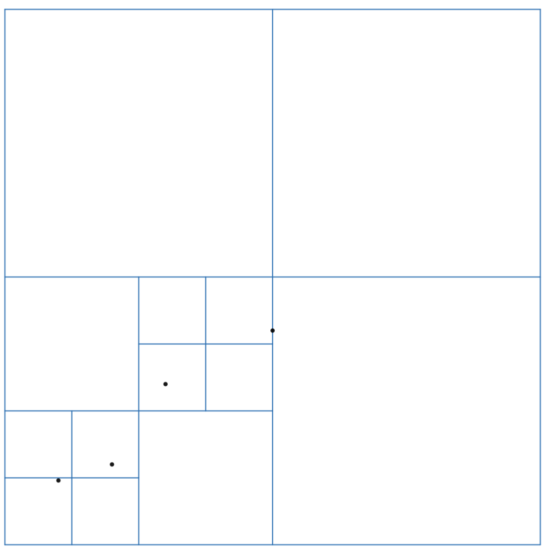
(20,15)



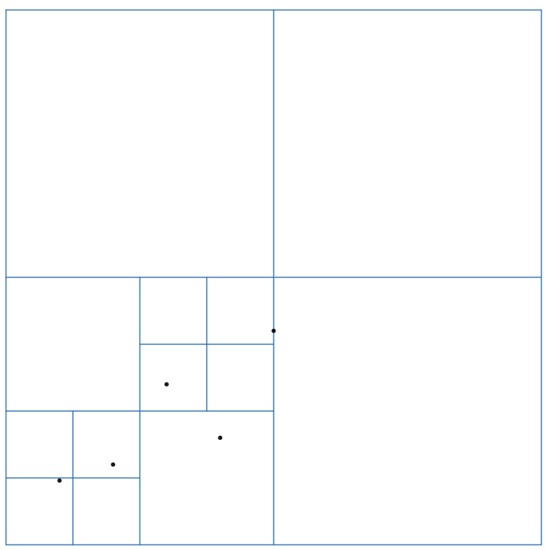
(50,40)



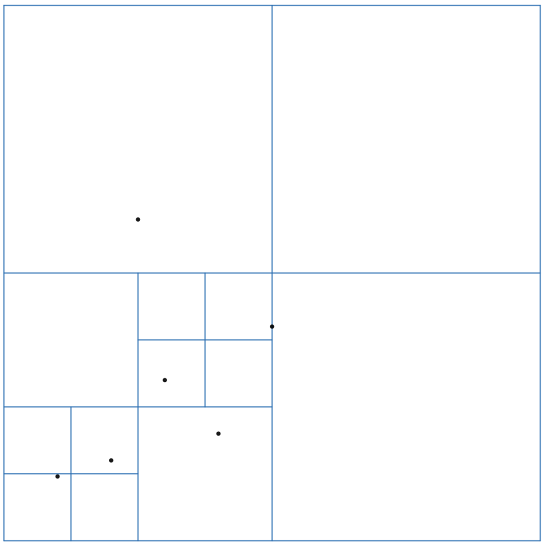
(10,12)



(40,20)



(25,60)



(15,25)