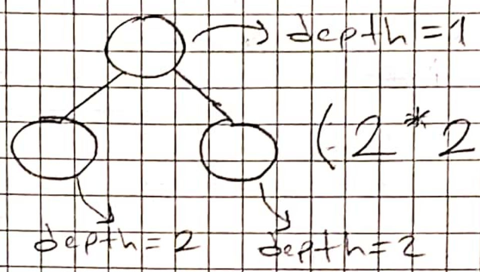
DERS :
KONU :

TARİH : ..../......./.......
GÜN :

1)

a) Calculate the total depth of nodes in a complete binary tree of height h.

if height is 2;



$(2*2)+(1*1)=5$

$$\text{depth in level}-1 = \begin{pmatrix}\text{height of}\\ \text{level}-1\end{pmatrix} * \begin{pmatrix}\text{number of nodes}\\ \text{at level}-1\end{pmatrix}$$

$$\Rightarrow h * 2^{h-1} = 1 * 2^{0} = 1$$

$$\text{depth in level}-2 = h * 2^{h-1} = 2 * 2^{1} = 2$$

$$\text{depth in level}-3 = h * 2^{h-1} = 3 * 2^{2} = 12$$

+

So, total number of depth in a tree of height 3 $= 12 + 2 + 1$

⇒ According to the above calculations, we can generalize formula:

$$\begin{pmatrix}\text{Total depth of}\\ \text{h height tree}\end{pmatrix} = \boxed{\left(2^{h-1}.h\right)+\left(2^{h-2}.(h-1)\right)+\ldots\ldots+\left(2^{0}.1\right)}$$

We can find total depth of nodes in complete binary tree of height h recursively;

In function int total_depth(int h);

1- if h=0 return 1 } its our base case

2- Else, return $(2^{h-1}.h)$ + total_depth(h-1).

* if we generalize the formula more with using geometric series, we'll find this; $\boxed{a(h)=(h-1).2^{h}+1}$

Scanned with CamScanner

DERS: Worst case = logn
KONU: Best case = 1

TARİH: ..../......./.......
GÜN :

b) Average number of comparisons in a BST

$$\text{Average number of Comparison in Binary Search Tree is} = \frac{(\text{Total number of Comparison})}{(\text{Number of Nodes})}$$

→ According to my research, total number of comparison has a same value with total depth of nodes.

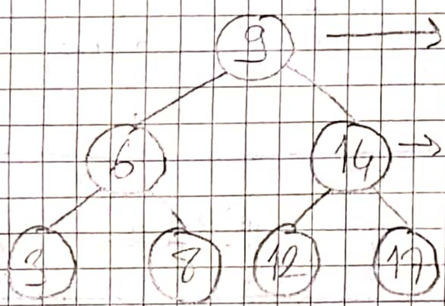
For height h; $n = 2^h - 1$

Total Number of Comparison $\Rightarrow a(h) = (h-1) \cdot 2^h + 1$

Number of Nodes $\Rightarrow n = 2^h - 1$

Average number of comparison is $\Rightarrow \dfrac{(h-1) \cdot 2^h + 1}{2^h - 1}$

Proof:



→ in level 1, there is 1 node and 1 comparison

→ in level 2, there is 2 nodes and 2 comparison

→ in level 3, there is 4 nodes and 3 comparison

So average number of comparison $= \dfrac{1*1 + 2*2 + 4*3}{7}$

$= 2.42$

With our formula (h=3);

Average num. of comparison $= \dfrac{(3-1) \cdot 2^3 + 1}{7} = \dfrac{2 \cdot 8 + 1}{7} = 2.42$

$2.4 = 2.4$, So formula is true.

Scanned with CamScanner

DERS :

KONU :

TARİH : ..../........../..........

GÜN :

C)

→ No, there is no restriction on the number of nodes in a full binary tree. Rule of being full binary tree is that each node has 2 or 0 children. It doesn't matter how many descendant they have.

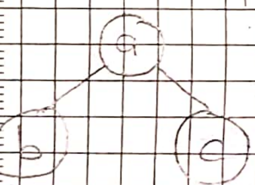
- A leaf node is node that has no children
- An internal node is node that has at least 1 child.

→ in full binary tree, nodes can have 0 or 2 children.

Lets start with 1 heighted binary tree.

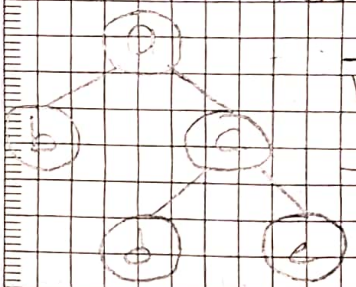
This tree has 1 leaf node and 0 internal node

If we add 2 children to node a;



→ Node a is no more leaf node but it created 2 more leaf node.

→ This tree has 2 leaf node and 1 internal node now. Lets add 2 children to node c.



→ Now, c become internal node, d and e become leaf node.

→ So, number of internal node and number of leaf node increased by 1. The difference between them doesn't changed.

→ Leaf node = 3, internal node = 2

→ Num of nodes = Num of leaf nodes + Num of internal nodes

⇒ We know that the number of internal nodes is always 1 less than number of leaf nodes.

⇒ So, if we say num. of internal nodes to k, k+1 became num of leaf nodes.

$$n = k + (k+1) \Rightarrow n = 2k+1 \Rightarrow \boxed{k = \frac{n-1}{2}}, \boxed{k+1 = \frac{n+1}{2}}$$

Num of internal nodes $= \frac{n-1}{2}$ //

Num of leaf nodes $= \frac{n+1}{2}$ //

DERS :

KONU :

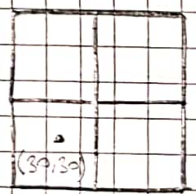
TARİH : ..../............/............

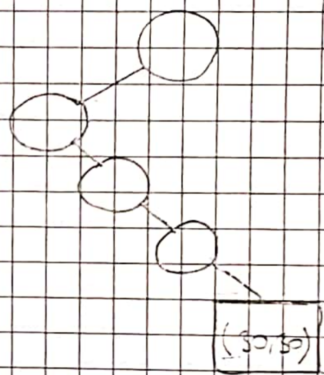
GÜN :

2- initial:

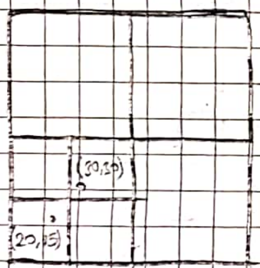
No Tree

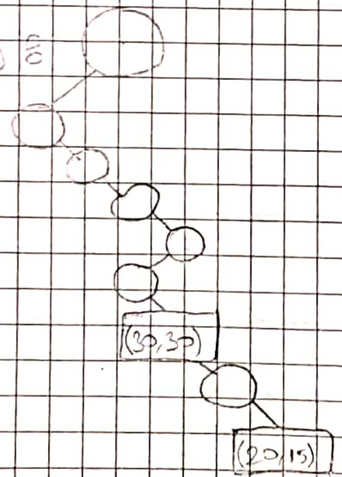
Inserting (30,30):



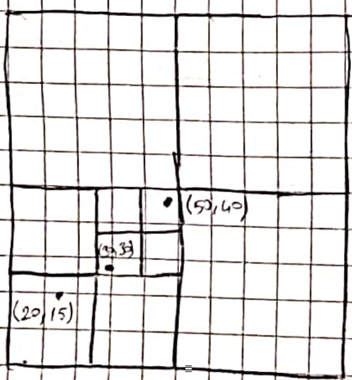


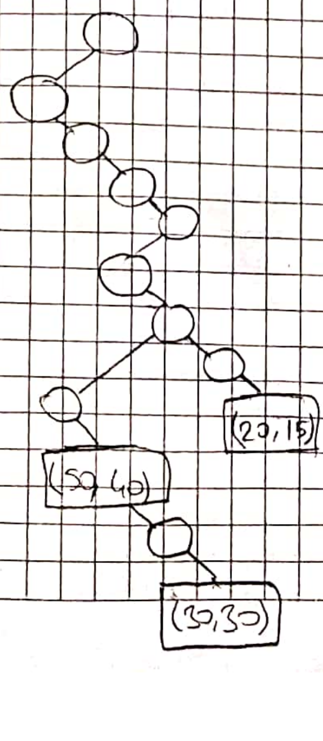
Inserting (20,15):





Inserting (50,40):





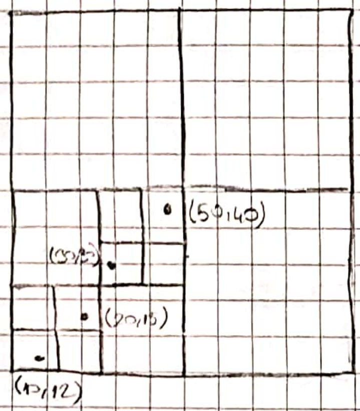
DERS :

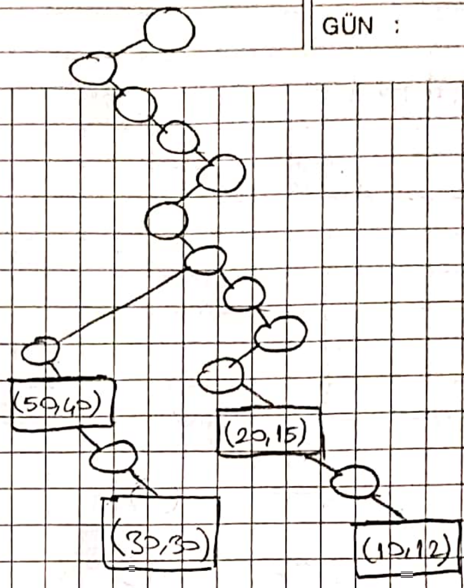
KONU :

TARİH : ..../........../..........

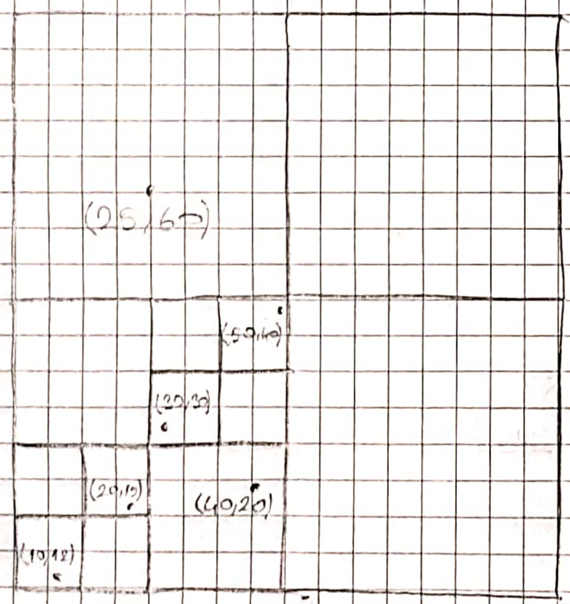
GÜN :

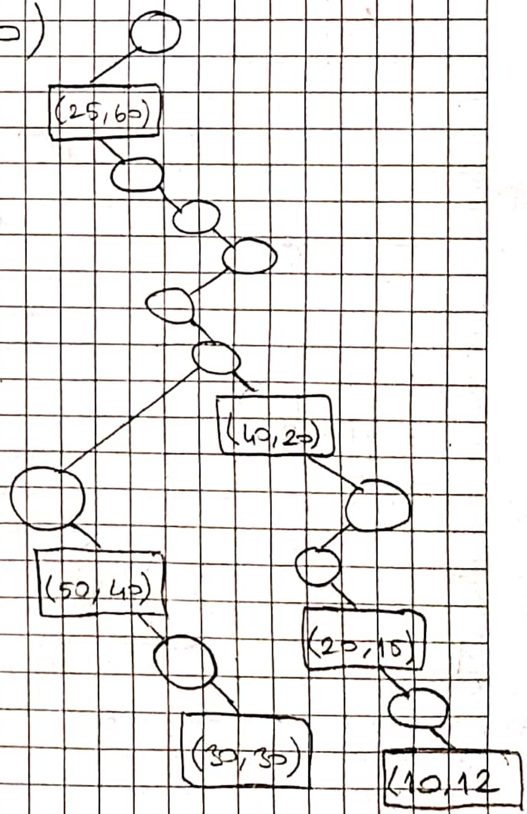
Inserting (10, 12) :





Inserting (40, 20) and (25, 60)





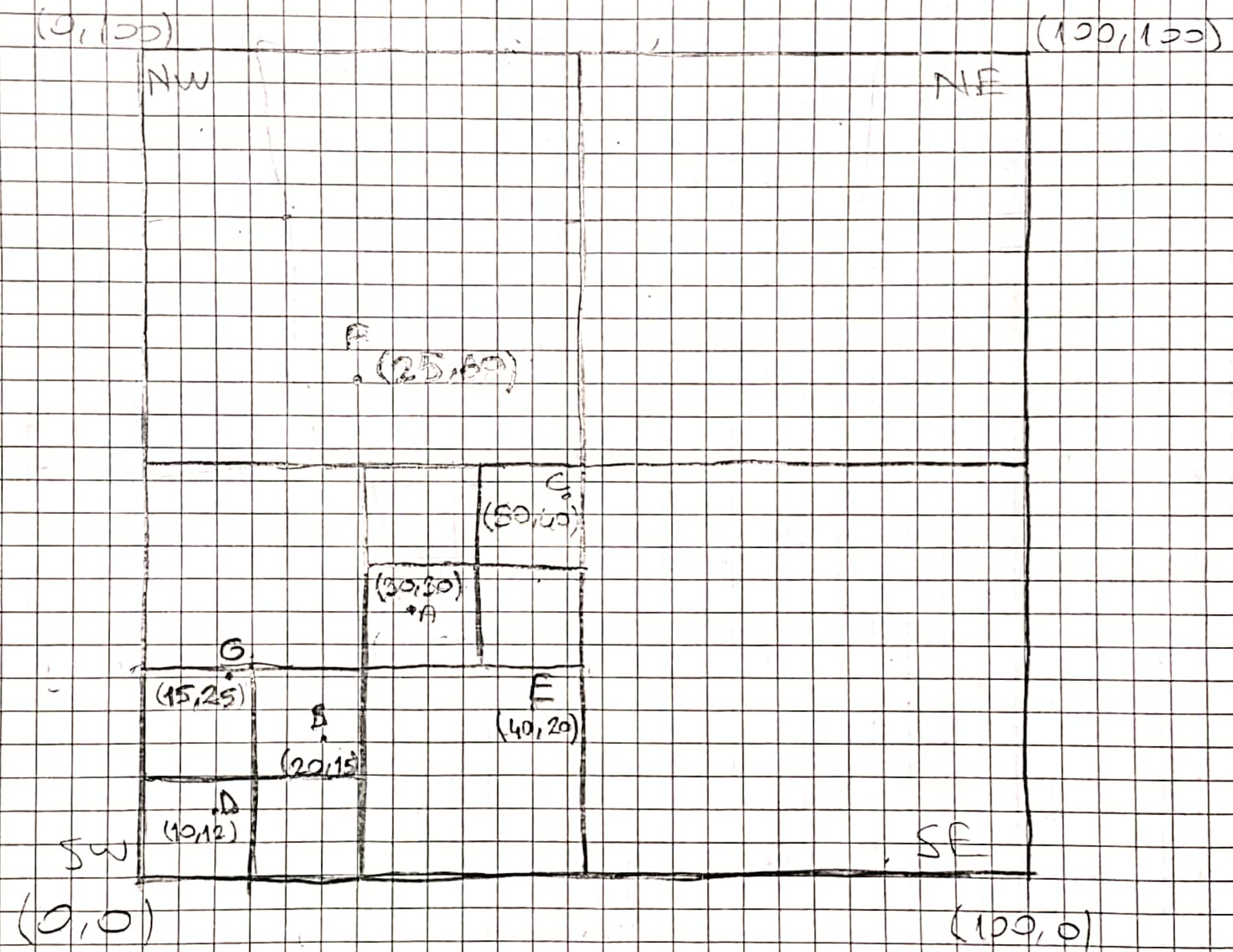
TARİH : ....../....../......

GÜN :

DERS :

KONU :

Inserting (15, 25) (Table)



DERS :

KONU : Inserting (15, 25) (Binary Tree)

TARİH : ...../...../.....

GÜN :