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

Homework
Report

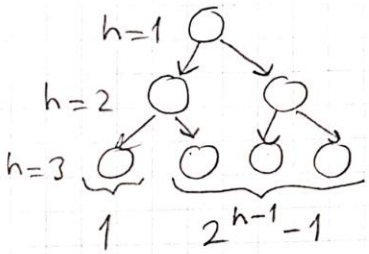
Sefa Çiçek

1801042657

1-

a)

height	depth
1	1
2	$1+(2\times2)$
3	$1+(2\times2)+(3\times2^2)$
4	$1+(2\times2)+(3\times2^2)+(4\times2^3)$



$\sum_{i=1}^{h} 2^{i-1} i = 2^h(h-1)+1 \Rightarrow$ This is for perfect binary tree. We must remove at least 1, at most $2^{h-1}-1$ node to obtain complete binary tree.

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

$Max = 2^h(h-1)+1-[(2^{h-1}-1).h]$

$Average = \frac{Min+Max}{2}$

$\frac{2[2^h(h-1)+1] \cancel{+h} - 2^{h-1}h \cancel{+h}}{2}$

$= \frac{2^{h+1}(h-1)+2-2^{h-1}h}{2} = \boxed{2^h(h-1)+1-2^{h-2}h}$

b)

Binary Search $\Rightarrow T(n) = T(n/2) + 1$

$T(n) = T(n/2) + 1$

$T(n/2) = T(n/2^2) + 1$

$T(n/2^2) = T(n/2^3) + 1$

$\rightarrow$

$T(n) = T(n/2) + 1$

$T(n) = T(n/2^2) + 1 + 1$

$T(n) = T(n/2^3) + 1 + 1 + 1$

$\vdots$

$T(n) = T(n/2^k) + k$

$n/2^k = 1 \Rightarrow n = 2^k$

$k = \log(n)$

$T(n) = \boxed{1 + \log(n)} \Rightarrow$ Average number of comparisons

c)

Restrictions on the number of nodes:

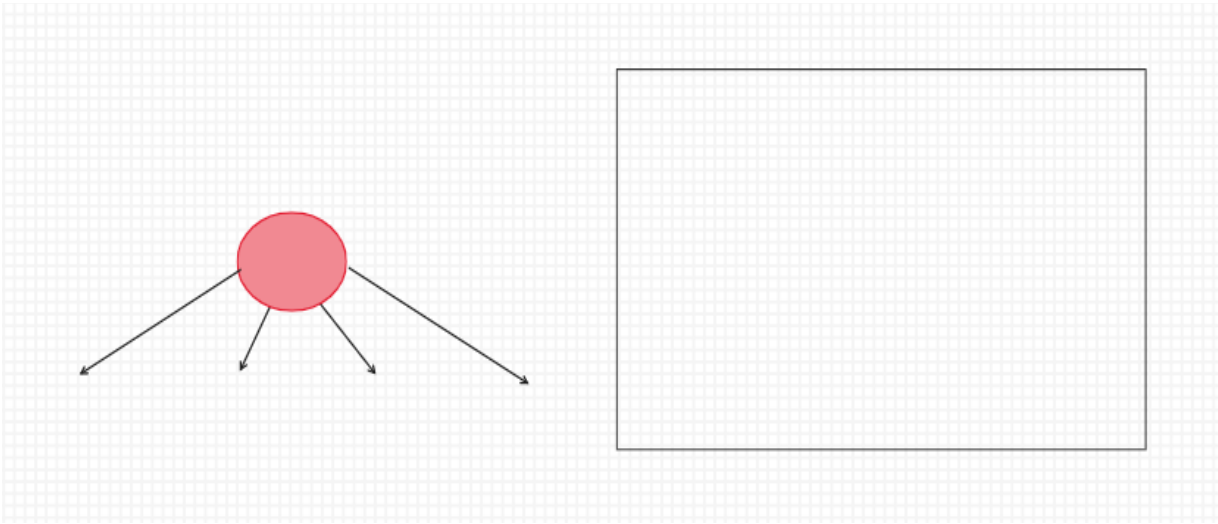
T : full binary tree .
* For every h ≥ 0, there are no more than 2^h nodes in level h.
* T (with h level) has no more than (2^h) – 1 nodes.

Number of internal nodes and number of leaves in an n node full binary tree:

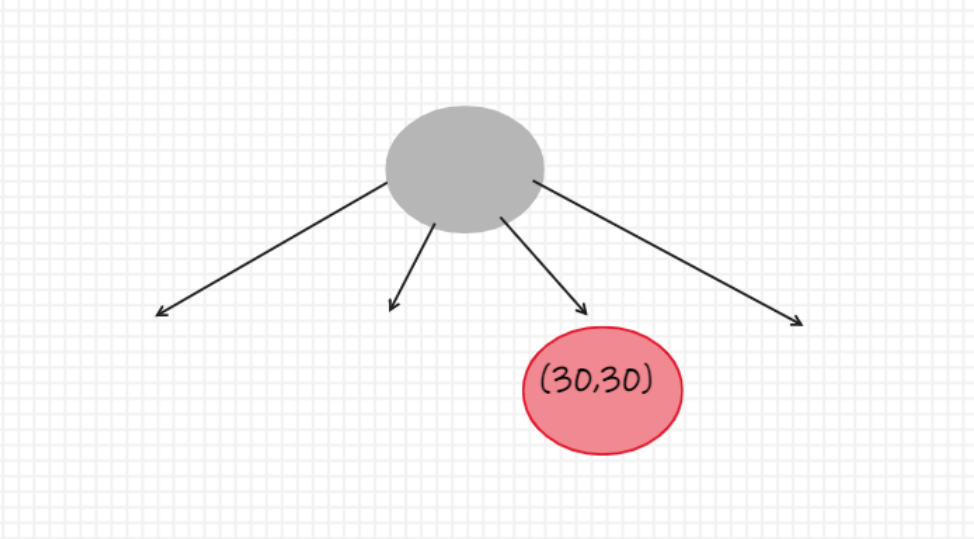
total number of nodes: N
number of leaves: L
number of internal nodes: I
- I = (N – 1)/2.
- L = (N + 1)/2.
- L = I + 1.
- I = L – 1.
- N = 2L – 1.
- N = 2I + 1.

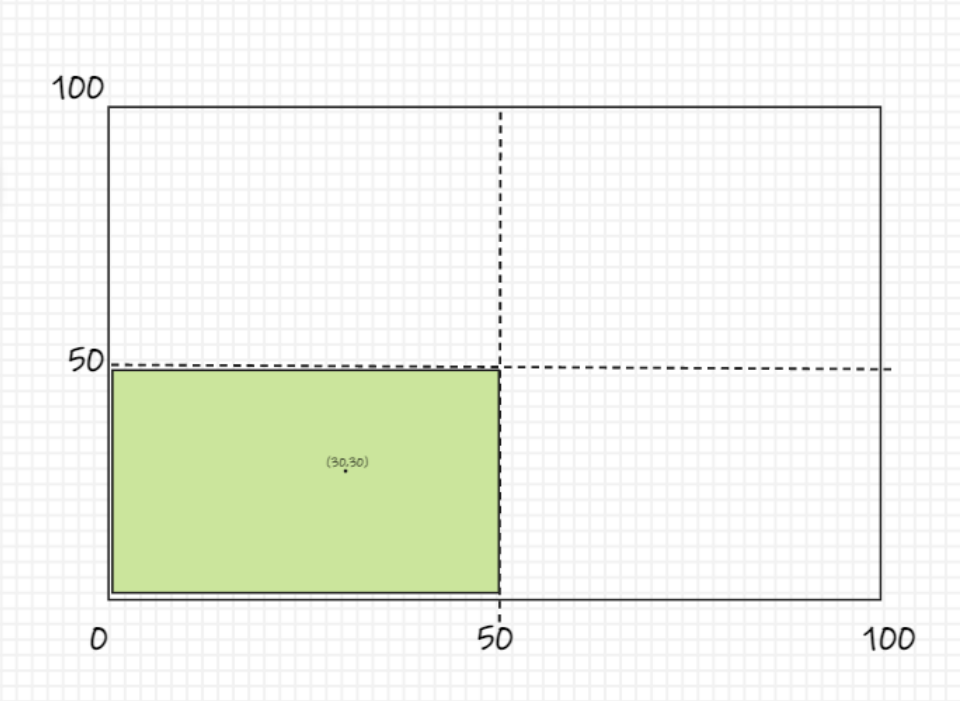
2- Quadtree

0) EMPTY TREE

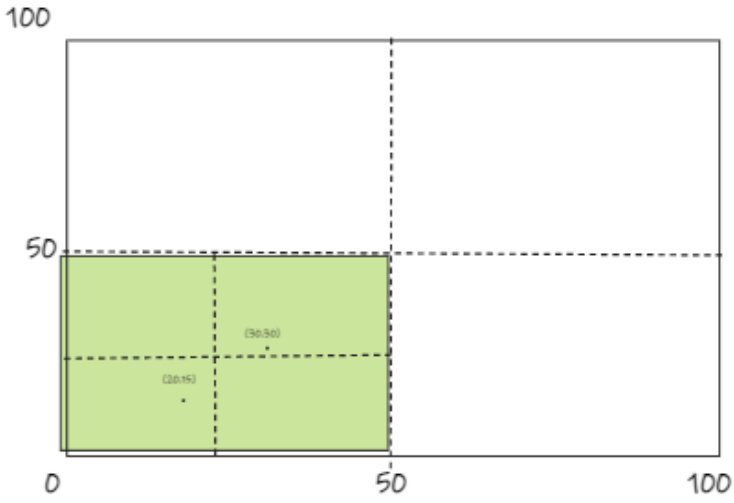


1) (30,30) has been added.

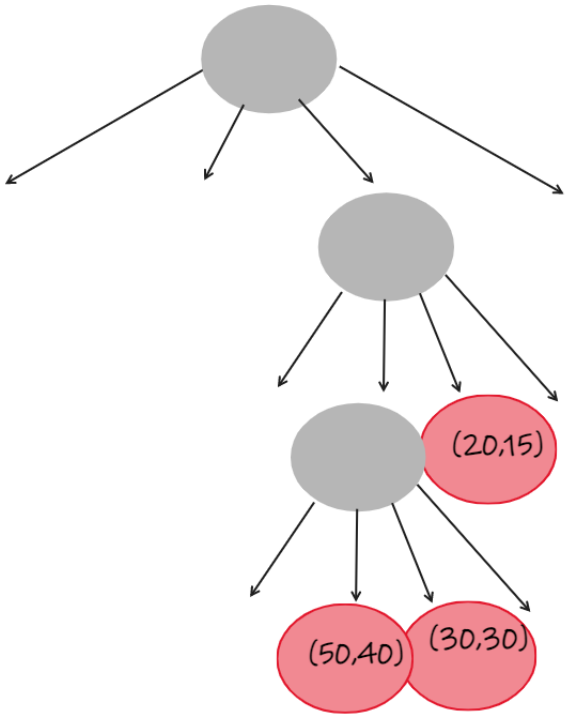


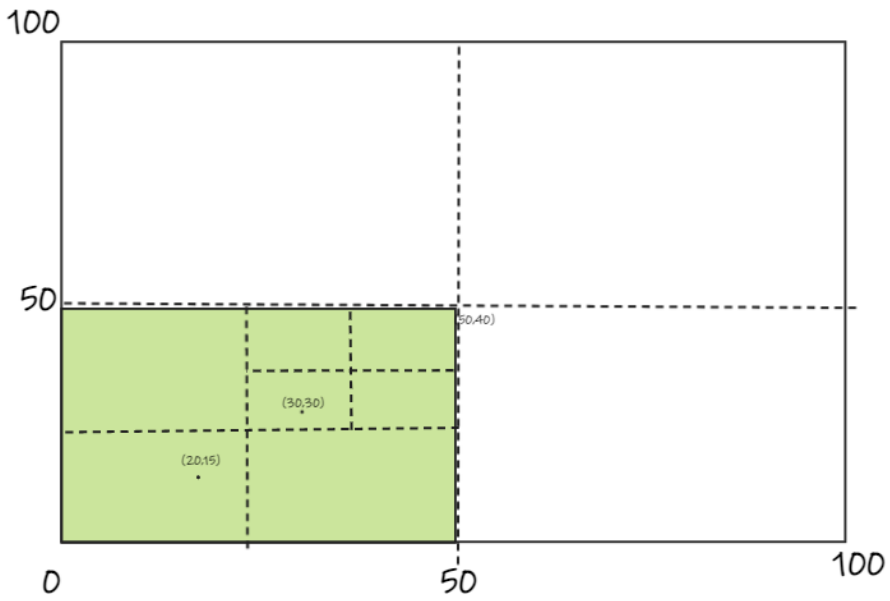


2) (20,15) has been added.

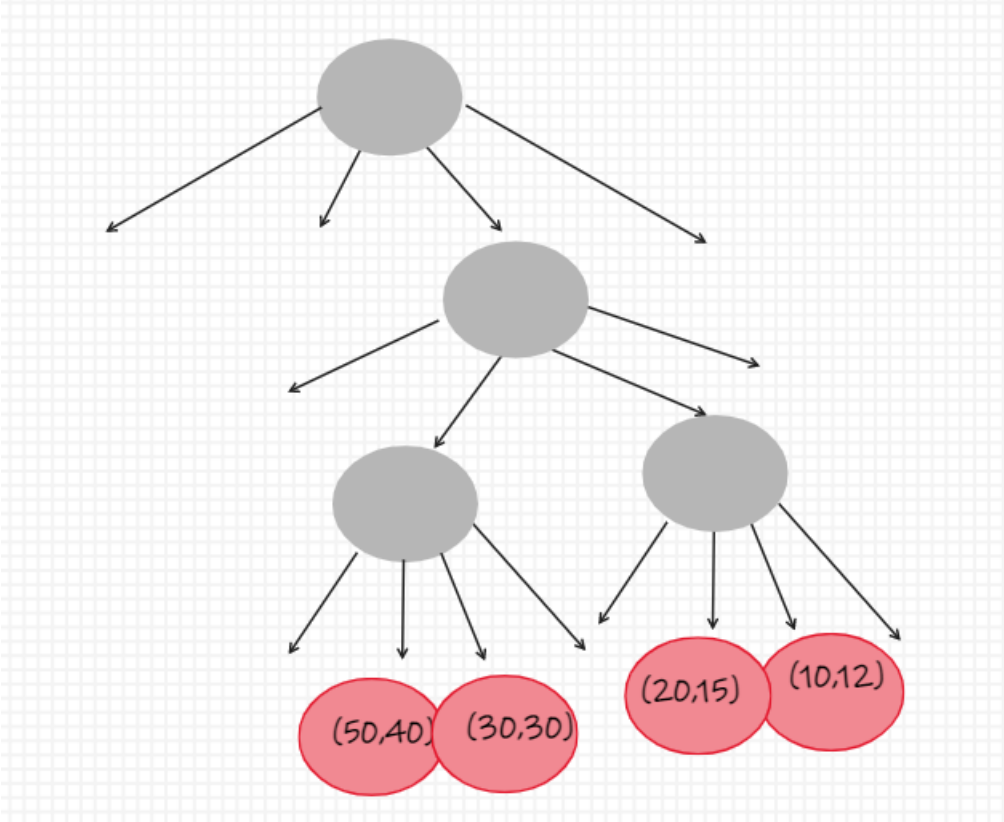


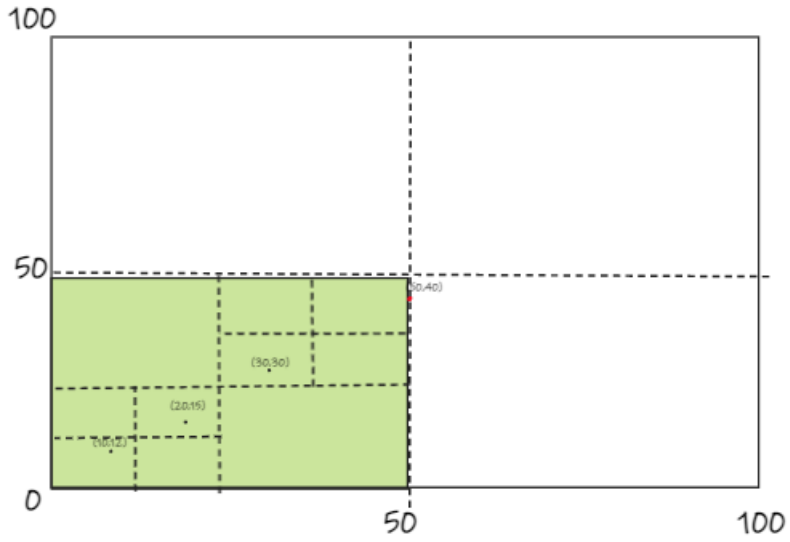
3) (50,40) has been added.



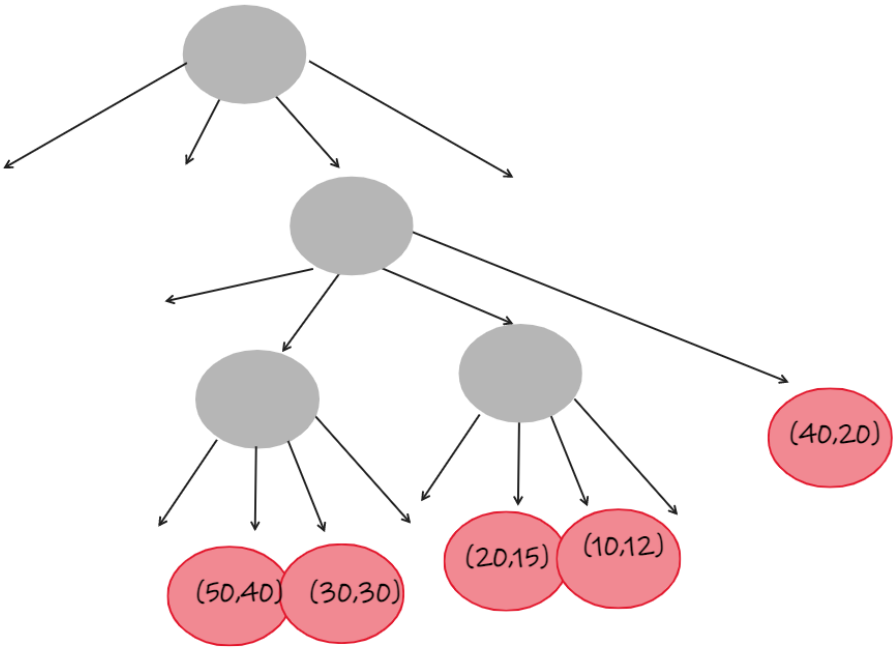


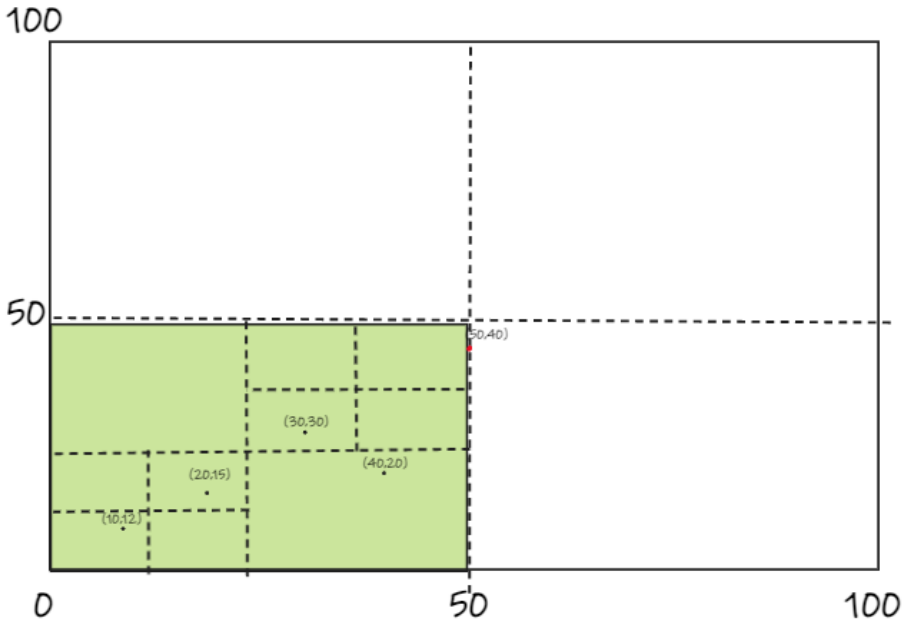
4) (10, 12) has been added.



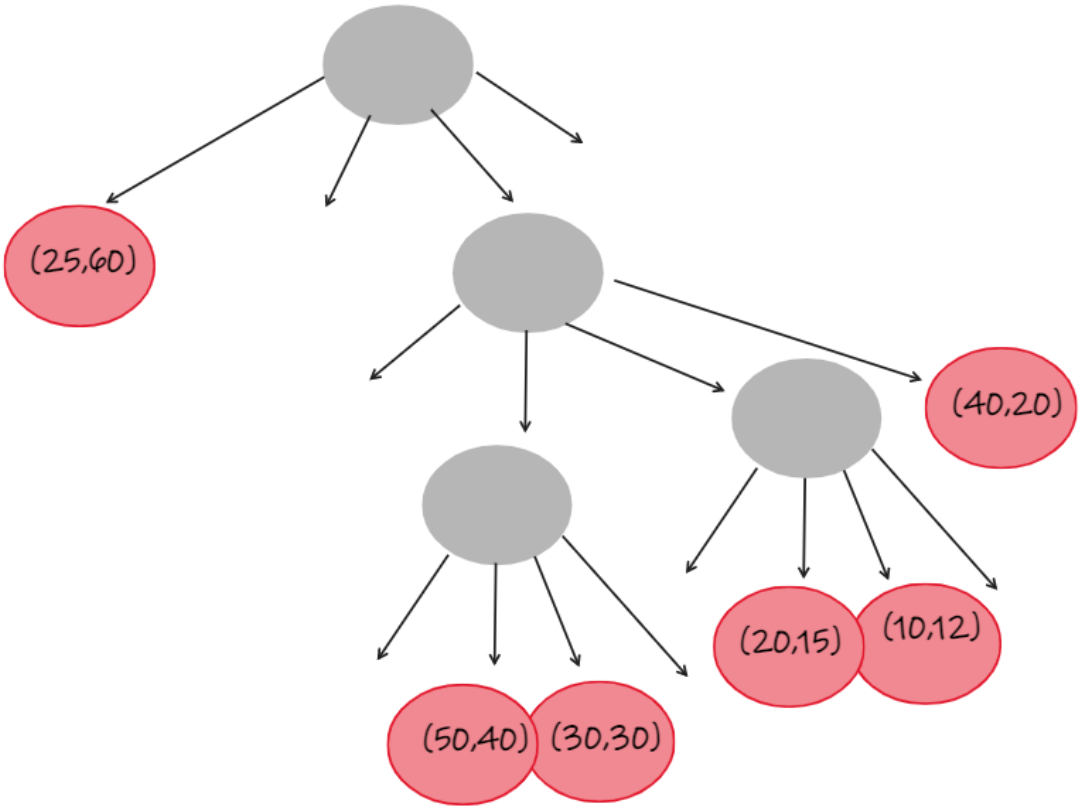


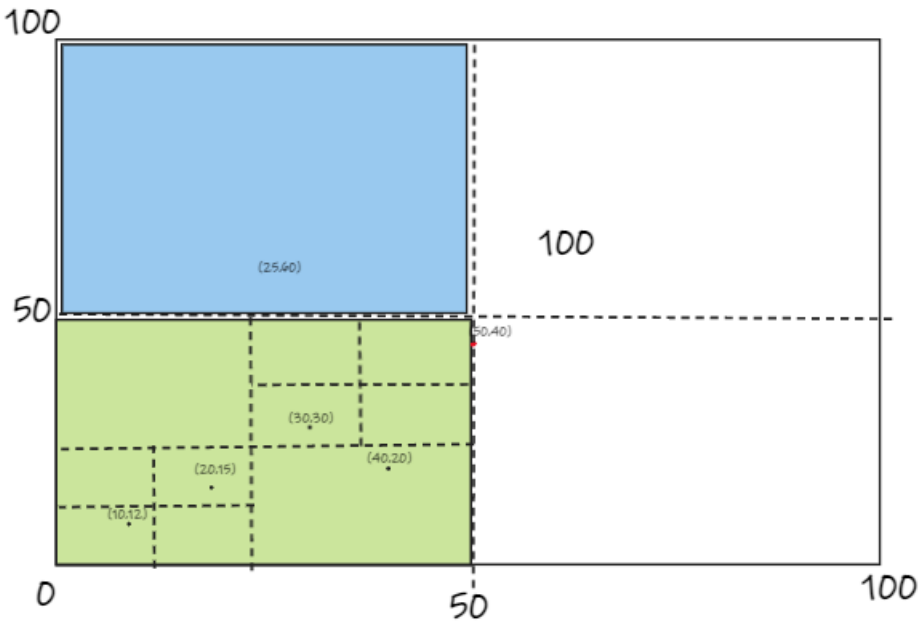
5) (40, 20) has been added.





6) (25, 60) has been added.





7) (15, 25) has been added.