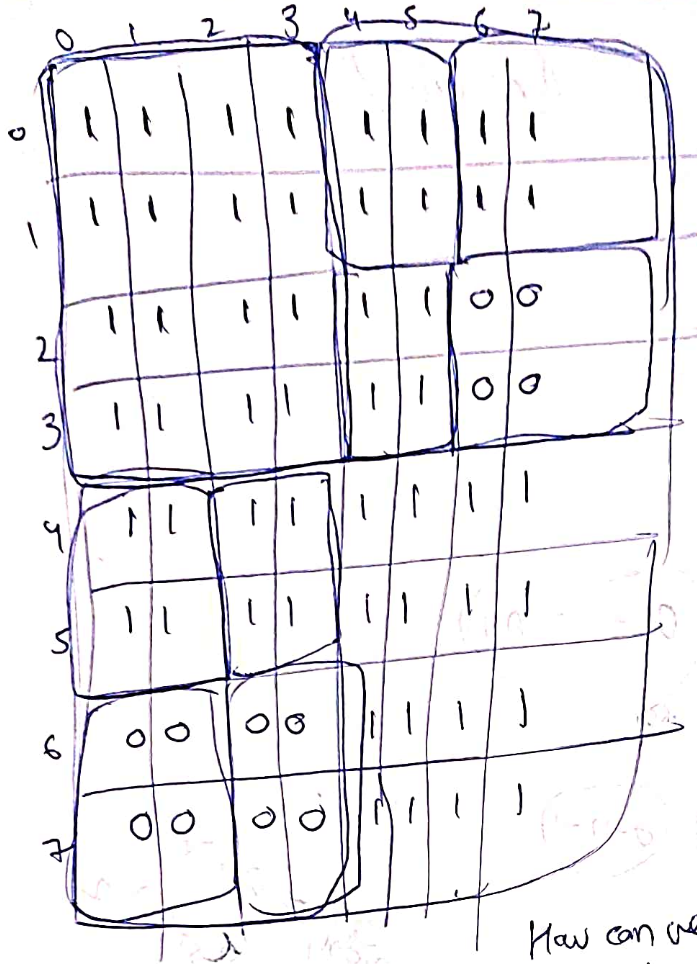
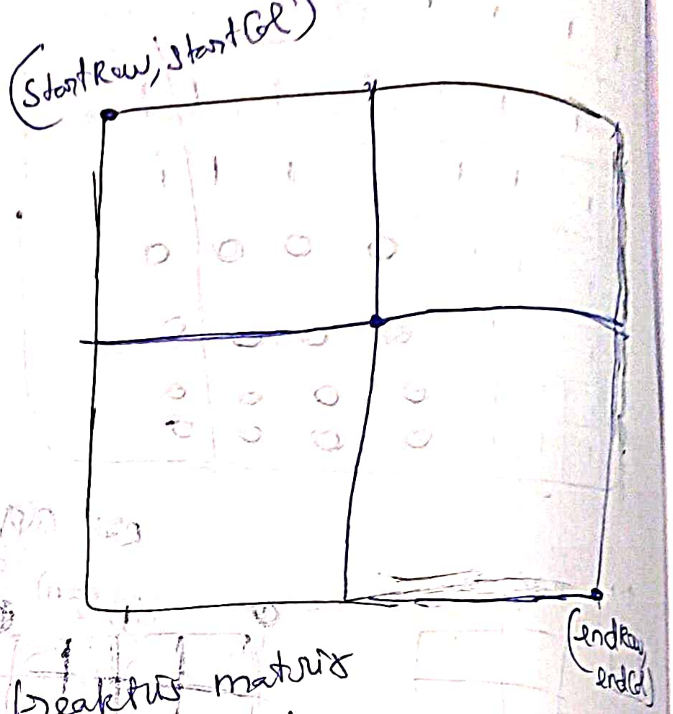


	0	1	2	3	4	5	6	7
0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	0	0
3	1	1	1	1	1	1	0	0
4	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1
6	0	0	0	0	1	1	1	1
7	0	0	0	0	1	1	1	1

thaught that came to my mind, find the nodes recursively

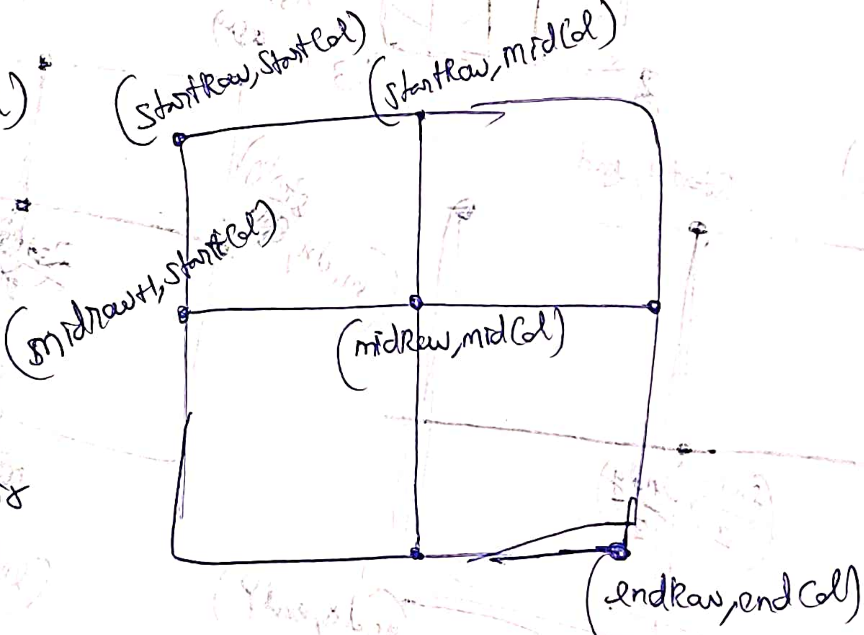


How can we breakthis matrix into multiple quadrants.

(startRow, startCol)

↓

(endRow, endCol)



Dividing the matrix into 4 quadrants

1st quad = (startRow, StarCol) → (midRow, midCol)

2nd quad = (startRow, midCol+1) → (midRow, end Col)

3rd quad = (midRow+1, startCol) → (endRow, midCol)

4th quad = (midRow+1, midCol+1) → (endRow, end Col)

this evaluation is fine

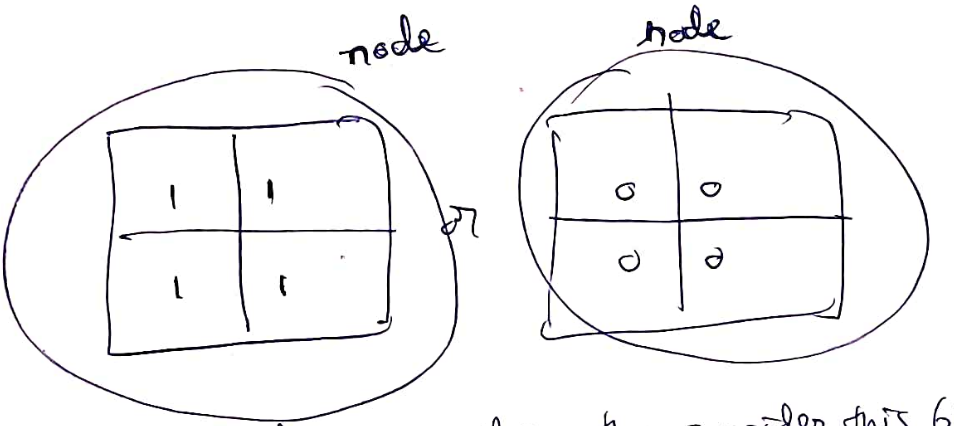
Mistakes that I did.

1st mistake

Considered only single cell as the leaf, rest of the nodes I did not consider as leaf.

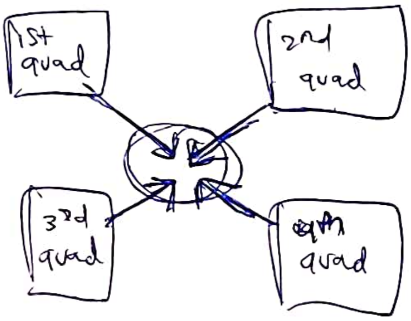
In my code all other nodes were not the leaf nodes.

whereas in the question, we had like



If all the nodes have the same value, then consider this big node as the leaf node.

2nd mistake



to consider a node as leaf I was only checking if their values are same,

And if values are not same, then we say node.isLeaf = false

node-val = any value 0 or 1

$\Longrightarrow$ So maybe. 1st quad is random val=1 and is not leaf, and other 3 are leaf and val=1, then node will be considered leaf.

Hence we need to check equal values as well as leaf node.