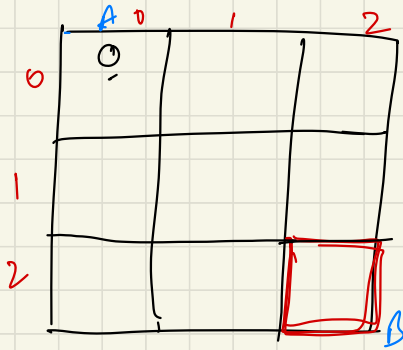
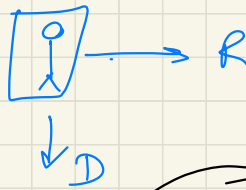



Q



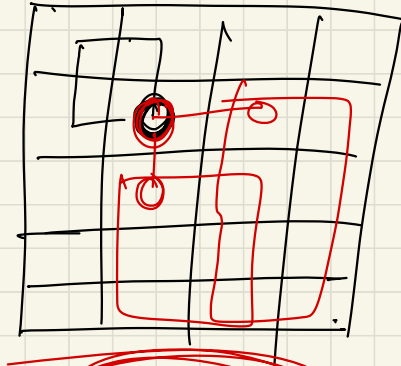
$(0,0) \rightarrow (2,2)$



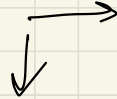
Ans:

RRDD
DDRR
RDDR
RDRD

(p, r, c)

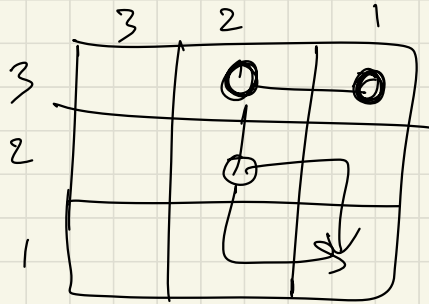
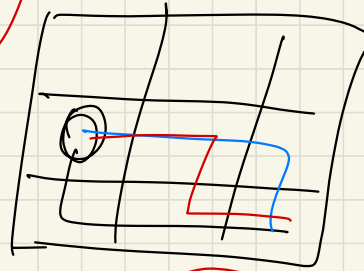
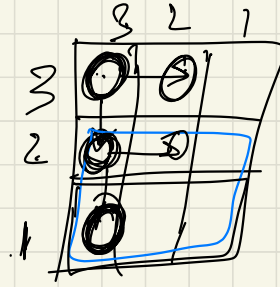
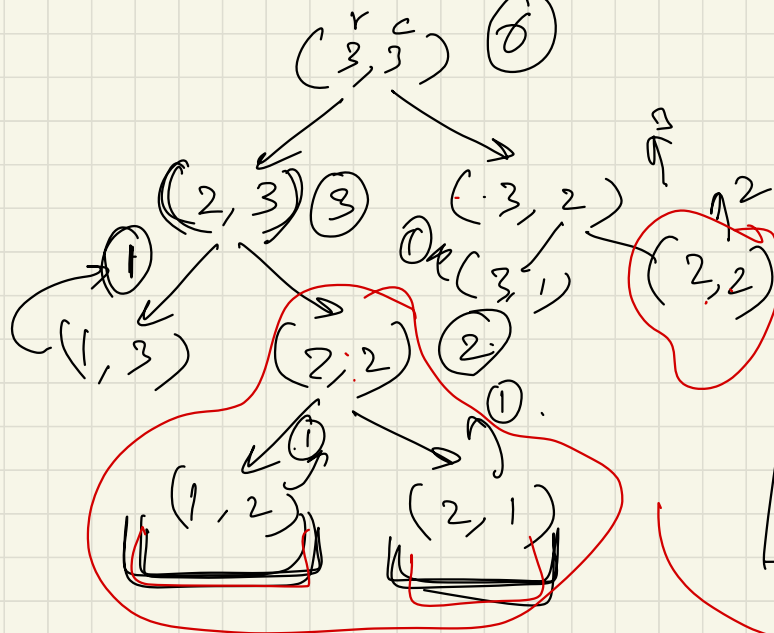


pick up



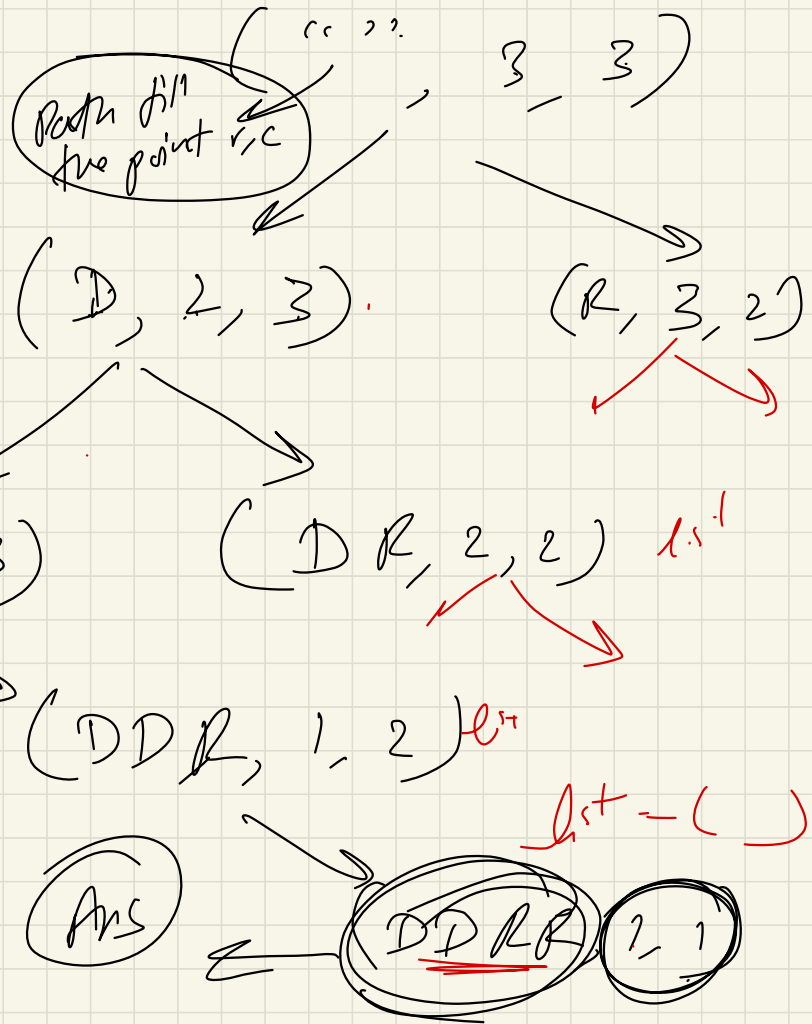
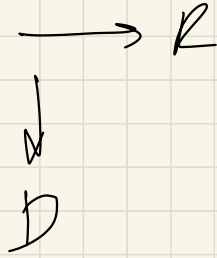
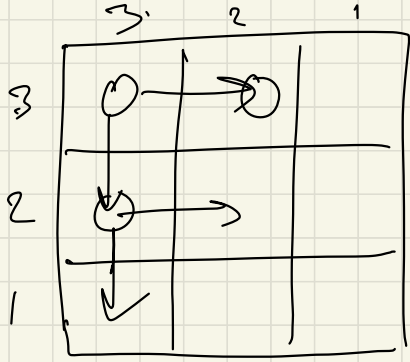
$RD + (\quad)$
 $DR + (\quad)$

\swarrow
(path till that point)

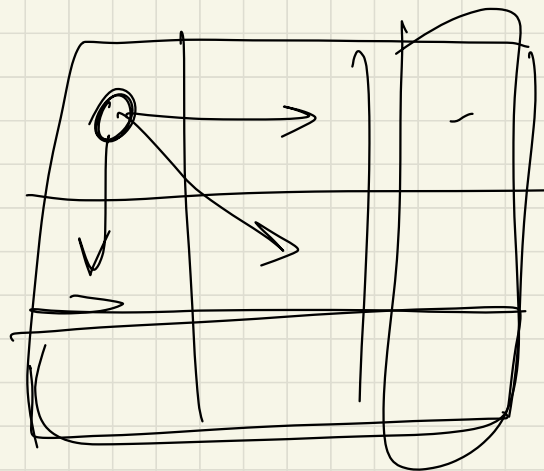


DP

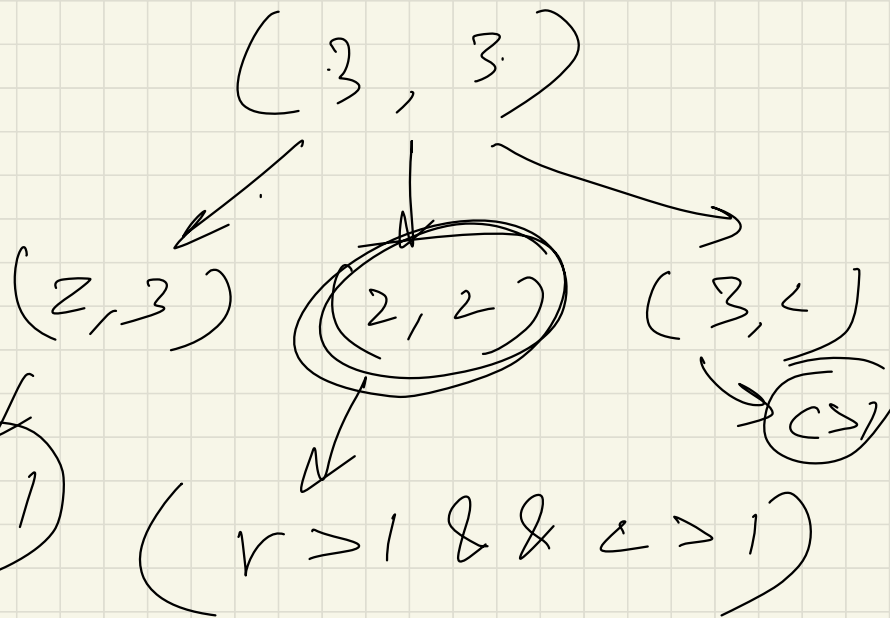
future
videos



Exactly same as subset p, up
 subset p, up
 p/s with 1 that
 vide. for complete
 decision. etc..

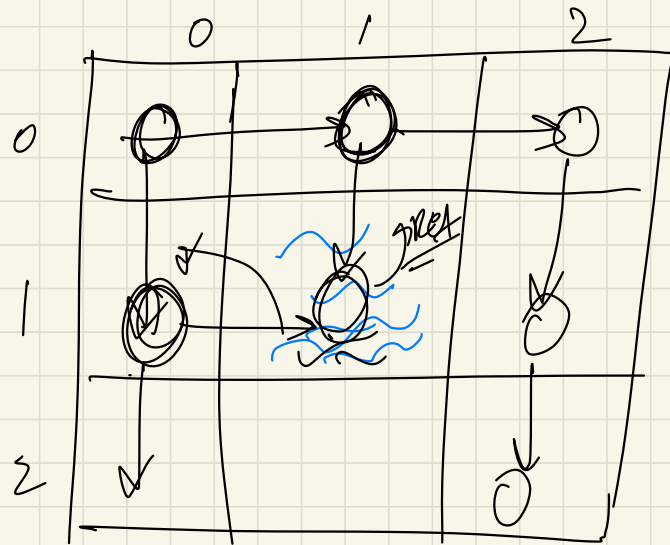


✓
 $(r > 1)$



Q:

Maze with obstacles:



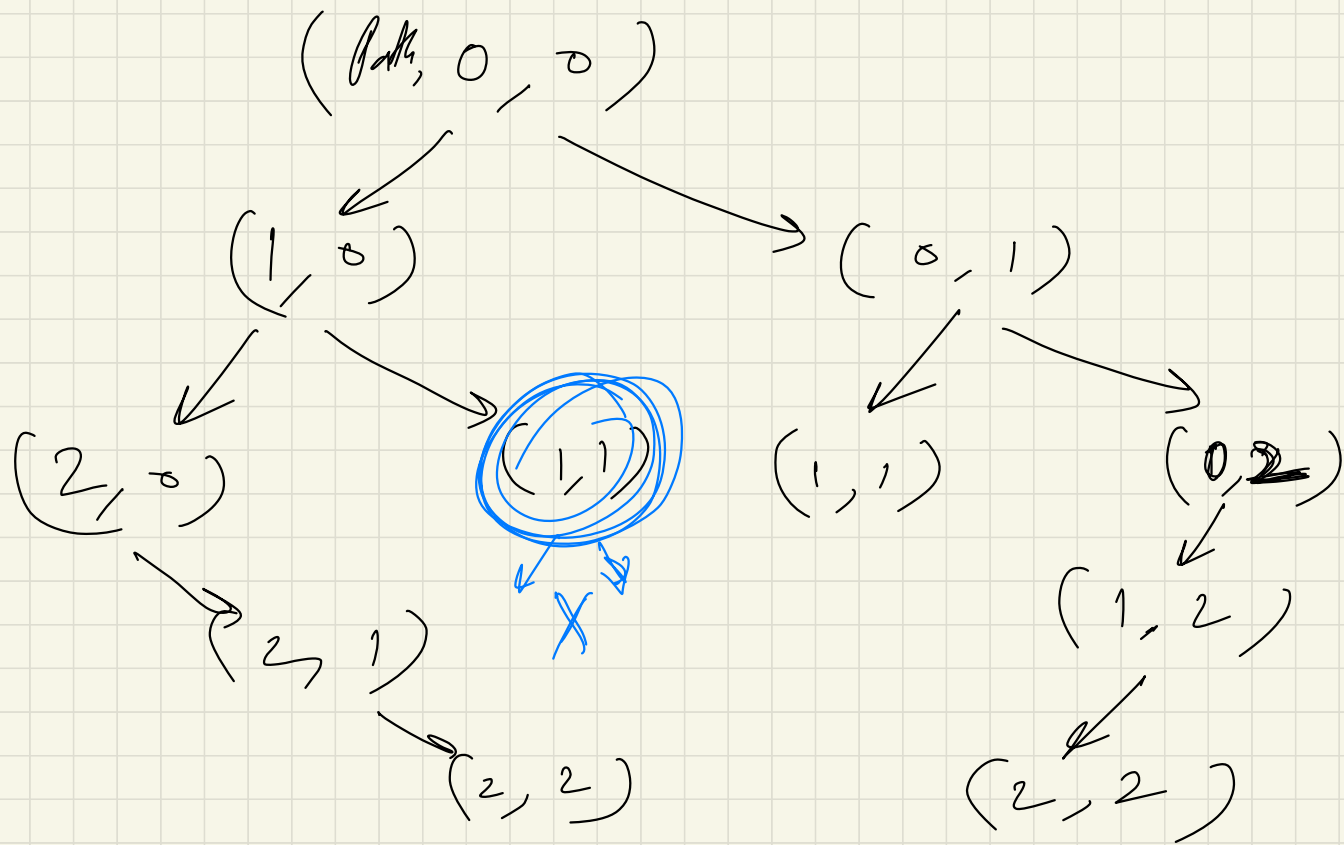
Boolean matrix

false → River

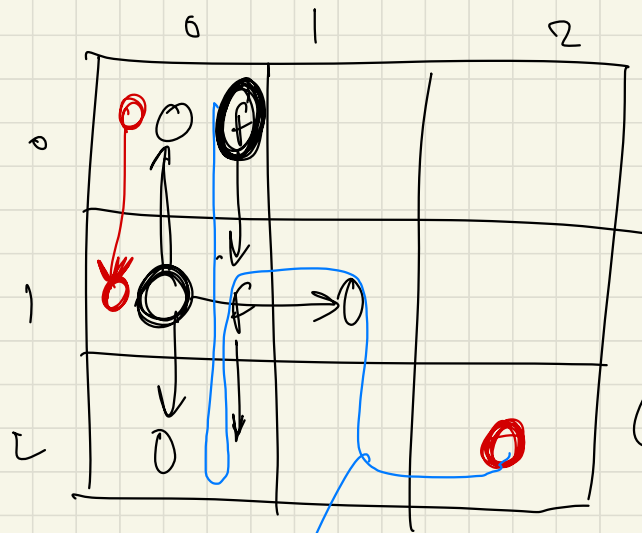
→ R
↓ D

Note: When you land on
a new cell,
check whether that
is river or not.

If you land on river,
stop recursion for that
cell.



Q:



Cannot be an answer:

Do not move back the same path

$(0,0,0)$

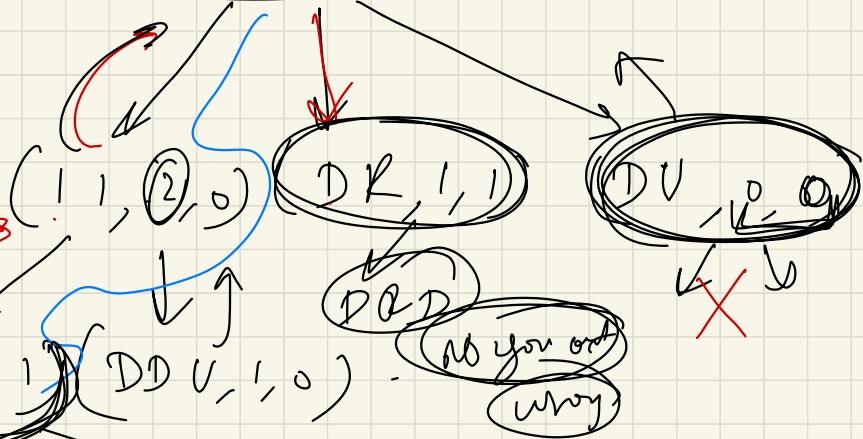
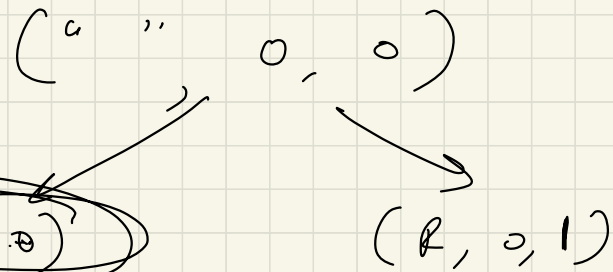
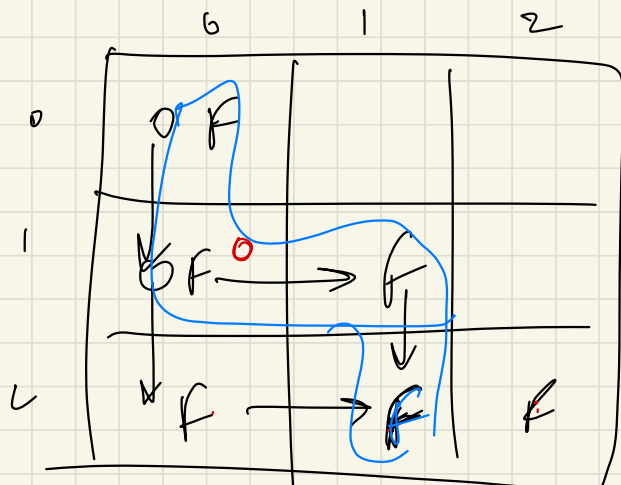
$(D,1,0)$ $(R,0,1)$

$(DD,2,0)$ $(DR,1,1)$ $(DV,0,0)$

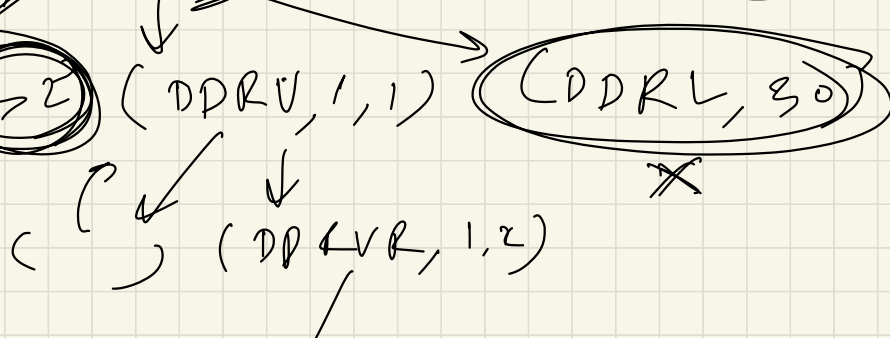
only new things

How to solve?

All cells that are visited, mark them as false. so that it does not go there.



happens when pr is not covered
 basic vec concept

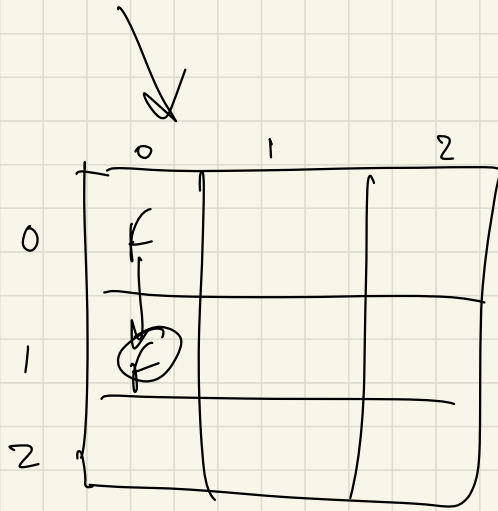


(DP LVR 1), 1, 2)

* Common sense: Marking false == I have that cell in my current path.

So when that path is over, ex: you are in another recursion call, these cells should not be false.

* While you are moving back, you restore the maze as it was.



This is known as
backtracking.

Very hard recursion ques!

When do we go back?

When the function is returned.

When you come out of the
recursive function \rightarrow you are
now in the same recursive call.
Hence, remember the cell as T.

Q:

1		
2		
3	4	

D D R R

1		
2	5	6
3	4	7

D D R V R D

- * Take a step variable
- * Update the path array
- * Print it in base condition
- * Backtrack