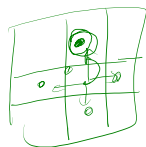
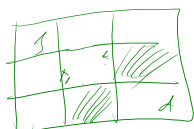


Q1 \rightarrow



facto \rightarrow force

Two \rightarrow block

Q2 \rightarrow maximum no. of moves to reach destination
minimum " " " " " " " "

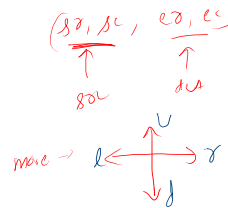
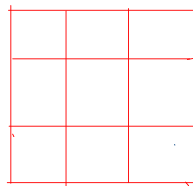
$$[(facto) \times 2] \times 2$$

$$[4k] \times 2$$

$$[4k, 4k]$$

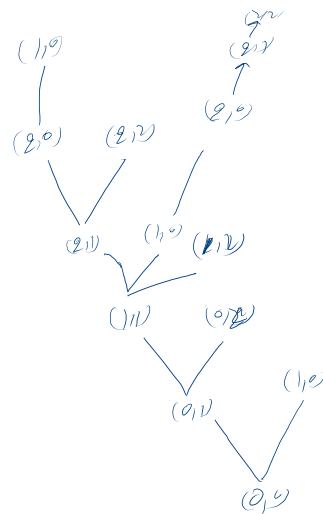
	0	1	2
0	.	.	.
1			
2			

val: 1



2) 0 \rightarrow

01

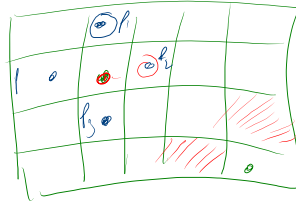


$$4k \rightarrow \boxed{4 \mid 4 \mid 4}$$

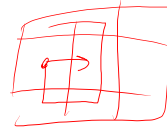
```

vis[sr][sc] = True
moves = 0
# for right movement
if sr + 0 <= er and sc + 1 <= ec and vis[sr + 0][sc + 1] == False:
    moves = max(moves, maximumMoves(sr + 0, sc + 1, er, ec, vis))
# for left movement
if sr + 0 <= er and sc - 1 >= 0 and vis[sr + 0][sc - 1] == False:
    moves = max(moves, maximumMoves(sr + 0, sc - 1, er, ec, vis))
# for down movement
if sr + 1 <= er and sc + 0 <= ec and vis[sr + 1][sc + 0] == False:
    moves = max(moves, maximumMoves(sr + 1, sc + 0, er, ec, vis))
# for up movement
if sr - 1 >= 0 and sc + 0 >= 0 and vis[sr - 1][sc + 0] == False:
    moves = max(moves, maximumMoves(sr - 1, sc + 0, er, ec, vis))
vis[sr][sc] = False
return moves

```



0 8 9 10 //

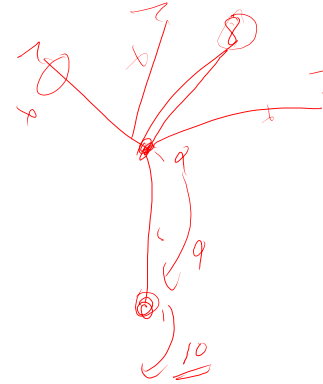
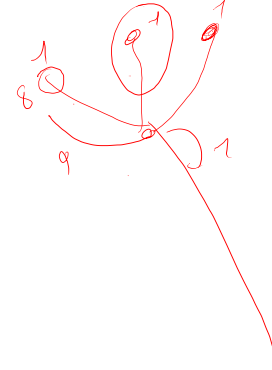
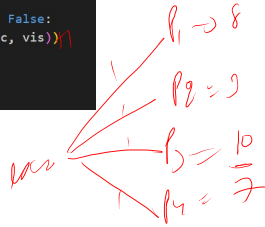


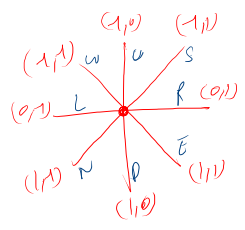
My locat: via p1 → 9

11 p2 → 9

11 p3 → 11

11 p4 → 8



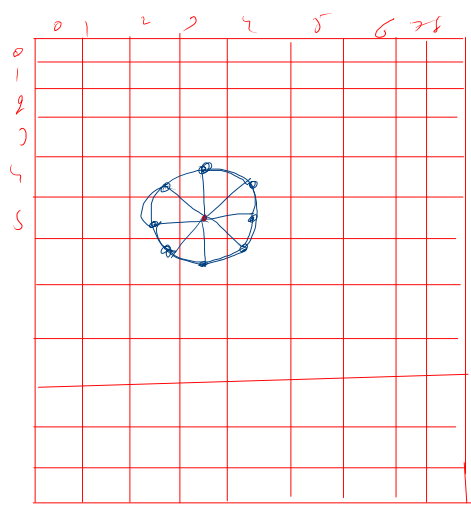
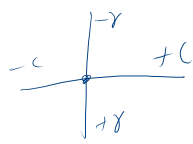


$(0 \leq d \leq 7)$
 $r = 80 + \text{dir}(d)(0)$
 $c = 80 \times \text{dir}(d)(1)$

dir

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

	U
1	S
2	R
3	E
4	D
5	N
6	L
7	W



```

def floodFill_02(sr, sc, er, ec, vis, dir, dirPath, ans):
    if sr == er and sc == ec:
        print(ans)
        return 1

    vis[sr][sc] = True
    count = 0

    for d in range(len(dir)):
        r = sr + dir[d][0]
        c = sc + dir[d][1]

        if r >= 0 and c >= 0 and r <= er and c <= ec and vis[r][c] == False:
            count += floodFill_02(r, c, er, ec, vis, dir, dirPath, ans)

    vis[sr][sc] = False
    return count

```

dir

0	1
0	1
1	1
2	0
3	0

dirPath

0	7
1	L
2	0
3	U