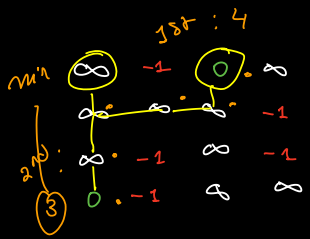
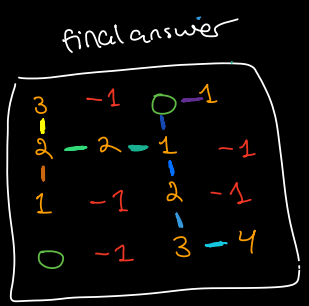
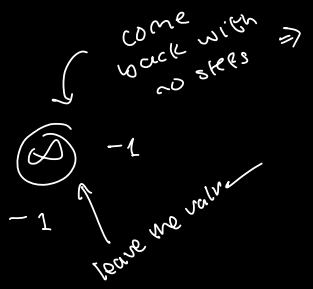
Problem: Given a 2d array containing -1: walls, 0: gates, INF: empty room
fill each room with the number of steps to the nearest gate

if it is impossible to reach a gate, leave INF as the value
INF = 2147483647







Intro:
- Verify Constraints
- Create Testcases

Brute Force:
- Brainstorming & Pattern Observations
- Pseudocode
- Write code
- Run through testcases
- Analyze time and space complexity

Optimal:
- Brainstorming & Pattern Observations
- Pseudocode
- Write code
- Run through testcases
- Analyze time and space complexity

start from gates and run dfs thru them

0 : [0, 2]
@ → and

try to fill in all positions distances

start dfs @ [0, 2]

sequencial search

if 0 :

start dfs @ mapping directory

0

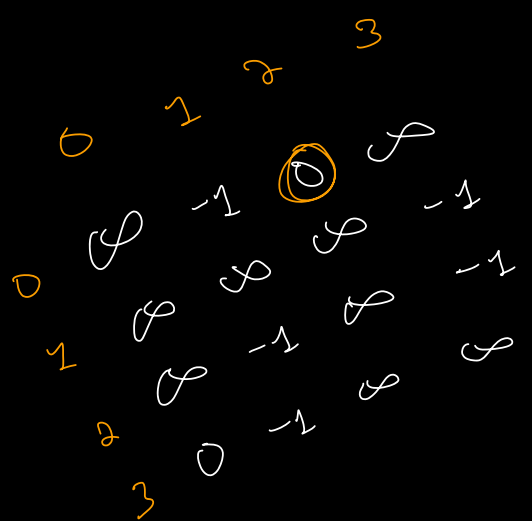
1

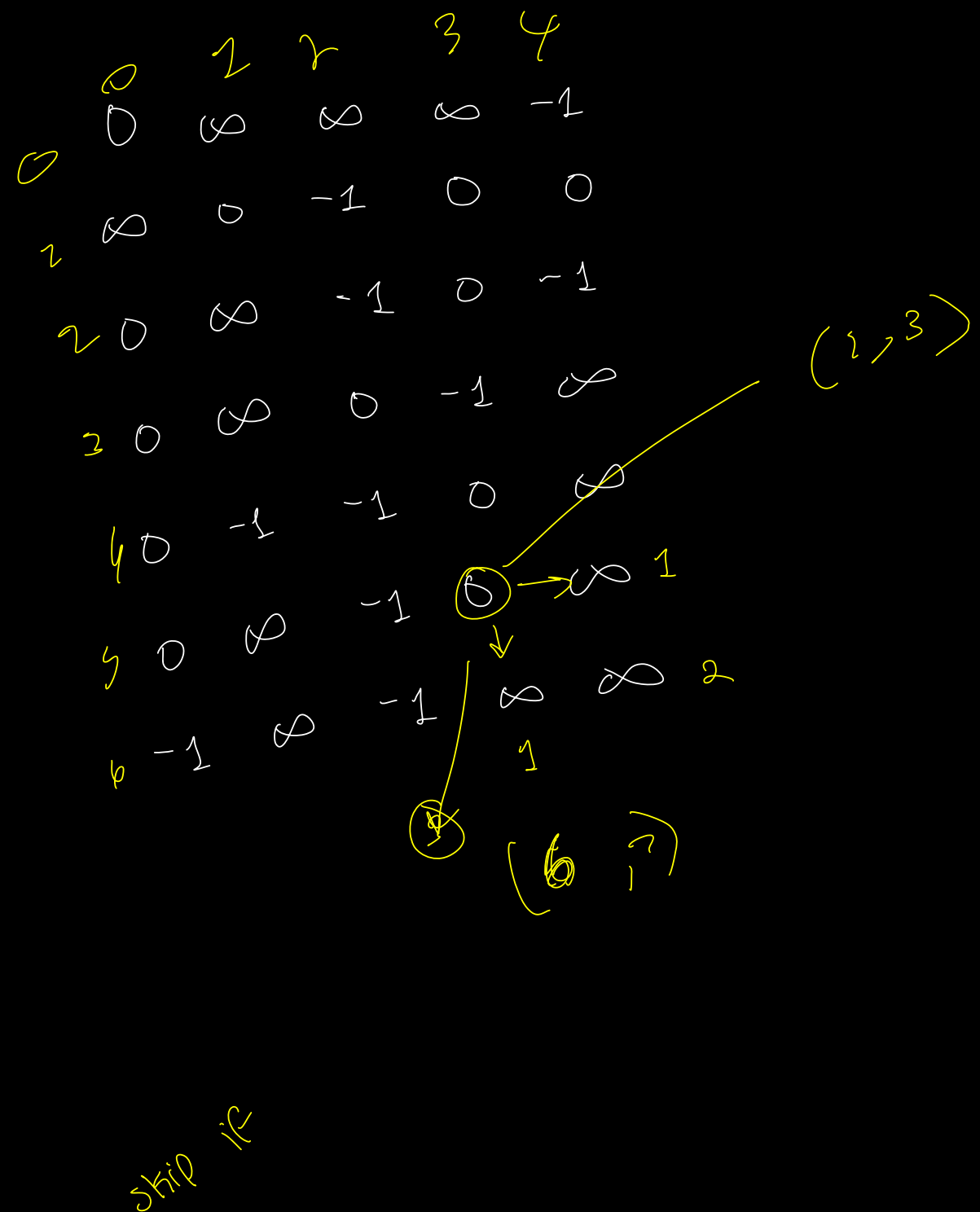
```
dfs (row, col, grid, stepsFromGate):

    for direction in directions:

        if row1 and col2 not in bound
                or
           grid[row1][col2] <= 0:
                continue

        else:
            stepsFromGate += 1
```





skip if