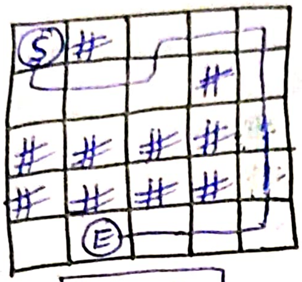
Graph Formulation 1

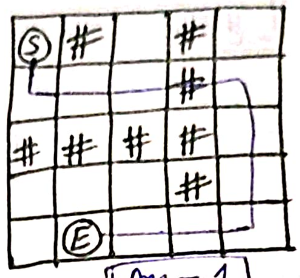
1. Given a grid with starting point and the ending point. Find :
Find the min. no of walls you need to break to go from pt S to E.

ⓐ Find the min. no of walls you need to break to go from pt S to E.



$\boxed{\text{Ans} = 0}$

No walls to break

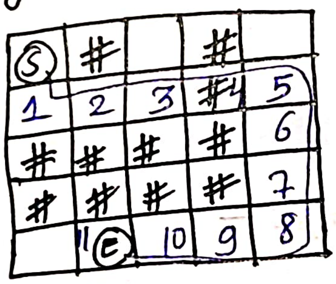


$\boxed{\text{Ans} = 1}$

1 wall needed to break

ⓑ If you can break <=k walls, Find min moves needed.

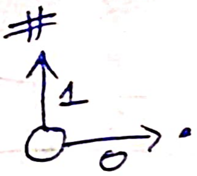
Exa:-



$K = 1$

$\boxed{\text{Ans} = 11}$

ⓒ Find all ans for ⓑ for k in range : $1 \le k \le N \cdot M$



Constraints : $N, M \le 100$.

a) While traversing the cell to go from S → E :

1. If cell is empty, then take cost = 0
2. If cell has a wall, then take cost = 1.

* Keep track of broken walls as when tracking these walls again will make cost = 0.

→ given code below : for printing path from S to E

(b) We have to minimise the no. of moves

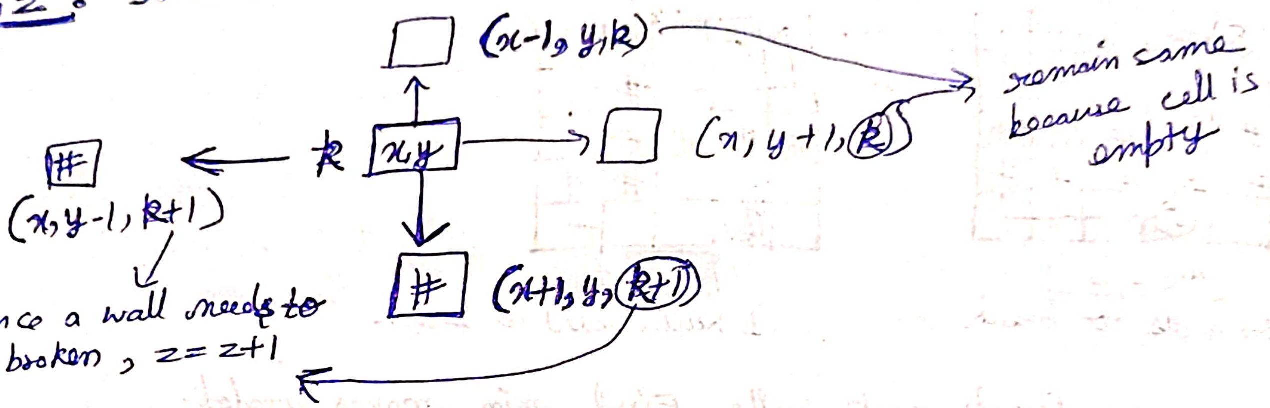
No. of walls broken <= k → Node (V)

Node is now (x, y, z). It will store

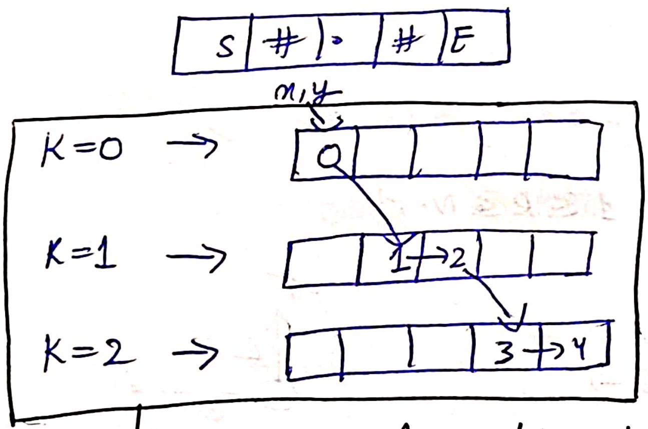
→ Min cost to (x, y) with z wall broken : This is our state.

Our real graph is now 3-D graph.

<u>x, y, z</u> : state at which we will run BFS Algo.

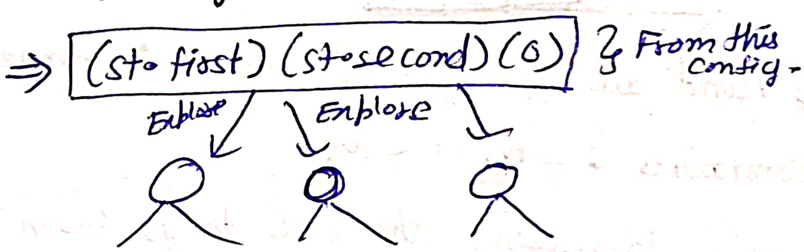


<u>suppose,</u>



This is the type of graph we're currently moving in.

Starting config :-



(E) ≤ k

After end point, ≤ k how many moves we can reach.

(c) Max. value of K = ~~N×M~~ N*M

Is N * M psbl ? No as for even the farthest pt, we can cover (N+M).

Use K constraint till N+M.

Code for (b) & (c) in live session chats