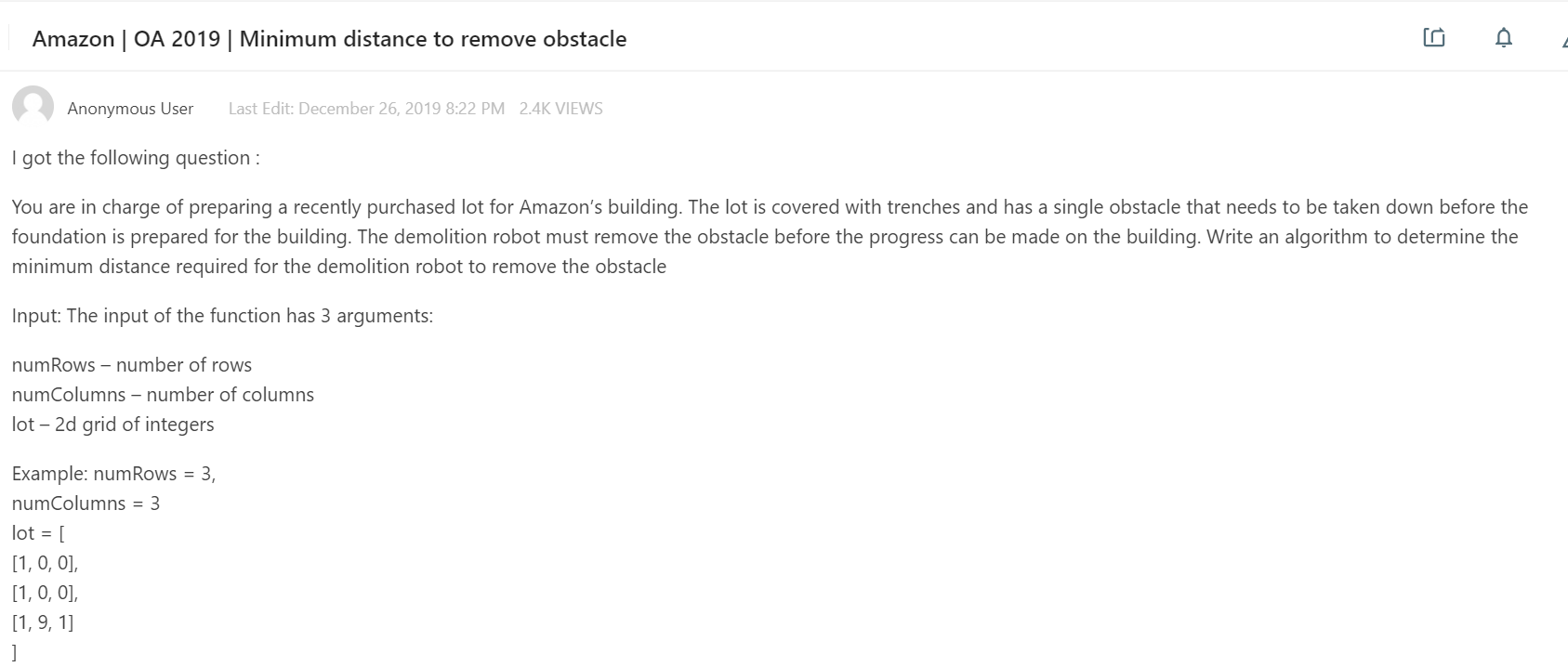
LEETCODE : <https://leetcode.com/discuss/interview-question/396858/amazon-oa-minimum-distance-to-remove-obstacle>

MINIMUM DISTANCE TO REMOVE OBSTACLES

https://leetcode.com/discuss/interview-question/396858/amazon-oa-minimum-distance-to-remove-obstacle



// "static void main" must be defined in a public class.

import java.util.ArrayList;

import java.util.LinkedList;

import java.util.List;

import java.util.Queue;

public class Impl {

public static void main(String[] args) {

List<Integer> ls1 = new ArrayList<>();

ls1.add(1);

ls1.add(0);

ls1.add(1);

ls1.add(0);

List<Integer> ls2 = new ArrayList<>();

ls2.add(1);

ls2.add(0);

ls2.add(0);

ls2.add(1);

List<Integer> ls3 = new ArrayList<>();

ls3.add(1);

ls3.add(1);

ls3.add(9);

ls3.add(0);

List<List<Integer>> list = new ArrayList<List<Integer>>();

list.add(ls1);

list.add(ls2);

list.add(ls3);

int rows = list.size();

int columns= list.get(0).size();

int shortestDistannce = getDistance(list,rows,columns);

System.out.println(shortestDistannce);

}

private static int getDistance(List<List<Integer>> list, int rows, int columns) {

// TODO Auto-generated method stub

boolean[][] visited = new boolean[rows][columns];

Queue<Node>q = new LinkedList<Node>();

q.add(new Node(0, 0, 0));

visited[0][0]=true;

while(!q.isEmpty())

{

Node top = q.poll();

visited[top.x][top.y]=true;

if(list.get(top.x).get(top.y)==9)

{

return top.dist-1;

}

if(top.x+1<list.size() && list.get(top.x+1).get(top.y)!=0 && !visited[top.x+1][top.y])

{

q.add(new Node(top.x+1,top.y,top.dist+1));

}

if(top.x-1>0 && list.get(top.x-1).get(top.y)!=0 && !visited[top.x-1][top.y])

{

q.add(new Node(top.x-1,top.y,top.dist+1));

}

if(top.y-1>0 && list.get(top.x).get(top.y-1)!=0 && !visited[top.x][top.y-1])

{

q.add(new Node(top.x,top.y-1,top.dist+1));

}

if(top.y+1<list.get(0).size() && list.get(top.x).get(top.y+1)!=0 && !visited[top.x][top.y+1])

{

q.add(new Node(top.x,top.y+1,top.dist+1));

}

}

return -1;

}

}

class Node {

public int x;

public int y;

public int dist;

Node(int x, int y, int dist) {

this.x = x;

this.y = y;

this.dist = dist;

}

}