[Robot Room Cleaner] (GOOGLE)

You have been given a robot and you have to use that robot to clean the room. The room has some of its cells blocked, and some cells are empty. ~~Your job is to say~~. You have been given the following APIs now you have to write an algo such that the blindfold robot cleans the entire room.

Robot → {

turnRight(); → turns the Robot to right by 90°.

turnLeft(); → turns the robot to left by 90°.

move(); ⇒ Move in the direction by 1 cell it is facing. returns false if there is a block, true if there isn't

clean(); → Cleans the current cell in which the robot is present.

}.

Soln 1) Idea is to solve it using dfs and backtracking. So basically we will traverse through the grid and check for all possible directions the Robot can travel and we will consequently clean the cell in which we are in.

2) The starting cell we will mark it as (0,0) and the remaining cells we will mark it relative to this cell.

3) Upon reaching a cell where all the options are blocked or we have already explored all the options we will backtrack to where we originally came from / [basically to a caller method] with

the same orientation. For that we will do the following operation.

turnRight() x 2 times → 180° to go in reverse.

move() → x1 time

turnRight() x 2 times → 180° to go regain original orientation.

directions → [{1,0}, {0,-1}, (-1,0), (0,1)]
Right, Down, Left, Upper

set<pair<int,int>> visited;

dfs(int x, int y, int current orientation){

visited[[(x,y)]] = 1

clean();

for(int i = 1; i <= 4; i++){

int new_orientation = (current_orientation + i) % 4;

int newX = x + directions[neworientation].first;

int newY = y + directions[neworientation].second;

turnRight();

if(!visited[newX][newY] and move()){ → Cell is empty

dfs(newX, newY, neworientation);

}

}.

turnRight();
turnRight();
move();
turnRight();
turnRight();
} → Explained in point 3)

Explanation.

Let's say
current orientation = 3.
i.e. Upper.
Robot is facing Up.
if we do
(Up+1 = 3+1 = 4) % 4
= 0.
it becomes Right.
So each time we are adding i we are basically turning Right()