## (GOOGLE) [Robot Room Cleaner]

Now howe been given a Tobot and you have to use that nobot to clean the Toom. The noom has some of its alls blocked, and some cells are empty. Your jet it to the You have been given the Robot of following APIs now you have to write an algo such that the blindfold Tobot cleans the entire Toom.

two Right (); -> two the Robot to rightly 40°.

two left (); -> two the Robot to left by 90°.

More (); > (move in the direction by real its facing.

Clean (); > Cleans the current cell in which the

70 bot is present.

(Soln) i) Idea is to solve it using dfs and back tracking. So busically we will proverse through the grid and check for all possity directions the Robot Can travel and we will consequently clean the cell in which we are in.

- 2) The storting all we will mark it as (0,0) and the remaining alls we will mark it relative to this all
- B) Upon reaching a cell where all the options one blocked or we have already explored all the options we will backtrack to where we originally came from [ basically to caller method] with

the same arientation. For that are will do the following operation.
tom Right () x 2times > 180° to go in Meroyl.
move () -> x1 time
torn Right () & 2 times > 180° to go regain original original original
Leavel of not booth about of special box il with
directions > \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
durections > [ { 1,0}, {0,-1}, [eft Urpor)  Set(power (int, int)) Usited 5
bounty ? hour - 21 301 Fill I way your 1 get show?
1. 101 Cana 1 . 1 . 1 .
olfs (int x, int y, int correct orientation)s  visited ((x,y)) =1  clean ();  Explanation.
Clean ();
Explanation.
for Cinti= 1-joi K= 9: ity was and a left Say
int new_orientation ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) (
int news - (conent overholion;)
int news - (conent overholion;)
int news - (conent overholion;)
int new y = y + directions [new crienteston]. flast   Robot is facing Up.
int new y = y + directions [ new crientestron) - flast   lare do    (1) Visited [ new 7]   gall is empty   (P+1 = 3+1=9)-/25.
int new $X = X + directions [ new crientation + i) = y - i.e. Oppen.  int new Y = Y + directions [ new crientation] - front  form Right()  [ Visited[new X][new Y] and move () ) }  Corrent others action ac$
int new $x = x + directions [neucriculation + i) = y - y - i.e. Oppen.  int new y = y + directions [neucriculation] - first   Robot is facing Up.  from Right().  [I we do  [I visited[new X][new Y] and move ()) }  dis(new X, new Y, new orientation).  for it fe comes Right.$
int new $x = x + directions [new crientation + i) = y - y - y - it. Oppen.  Int new y = y + directions [new crientation] - flat,  then Right().  If we do  (Up t = 3 + (= y) = y - y  dys (ments, new y, new crientation);  So lach time-we are  So lach time-we are$
int new $X = X + directions [ neutrinatation + i) / o y [il. Opper.]  Int new Y = Y + directions [ neutrinatation] \cdot floor)  Robot is facing Up.  Robot is facing Up.  If we do  VP+(=, 3+(=9)=/o y  dis(ment) new Y new orientation);  So lach time we are  adding i we one  adding i we one$
int new $X = X + \text{directions} \left[ \text{new crientation} + i \right] /_{0} \text{ y} \right]$ int new $Y = Y + \text{directions} \left[ \text{new crientation} \right] \cdot \text{first} \right]$ Robot is facing Up.  If we do  If ( ! visited [ new X] [ new crientation ] - Fread:  Office ( new X) [ new Y] and move ( ) ) S  If if fe comes Right.  So lach time we are  adding i we one
int new $X = X + \text{directions} \left( \text{new riendation} + i \right) /_{0} y_{5}$ int new $Y = Y + \text{directions} \left( \text{new riendation} \right) \cdot front$ Robot is facing Up.  How do  (! voited (new X) (new Y) and move ()) }  dels (new X) (new Y), new orientation);  turn Right ();  furn Right ();  furn Right ();  furn Right ();
int new $X = X + directions [new crientation] + i) + i + i + i + directions [new crientation] + i) + i + i + directions [new crientation] + i + i + i + i + i + i + i + i + i + $
int new $X = X + \text{directions} \left( \text{new riendation} + i \right) /_{0} y_{5}$ int new $Y = Y + \text{directions} \left( \text{new riendation} \right) \cdot front$ Robot is facing Up.  How do  (! voited (new X) (new Y) and move ()) }  dels (new X) (new Y), new orientation);  turn Right ();  furn Right ();  furn Right ();  furn Right ();