

J. CLEANER ROBOT

PROBLEM DESCRIPTION

給一個方格地圖，機器人的位置及面向方向，機器人會不斷清理及移動位置，遇到前方死路會右轉，問最後機器人會清理多少面積。

SOLUTION TECHNIQUES

BFS

SOLUTION SKETCHES

很經典的廣度優先搜索題目。

使用 `queue` 並記錄走過的位置、走過的方格數量。

BFS 的技巧就是宣告方向向量如

```
int mx[4] = {-1, 0, 1, 0}, my[4] = {0, 1, 0, -1}
```

一次即可做好四個方向，同時對於轉 90 度也會輕而易舉。

TIME COMPLEXITY

$O(W * H)$ ， W 為地圖的寬度、 H 為地圖的高度。

SOLUTION PROGRAM FOR REFERENCE

```
#include <iostream>
#include <cstdio>
#include <cstring>
using namespace std;
char g[15][15];
int mx[] = {-1,0,1,0};
int my[] = {0,1,0,-1};
bool vst[10][10][4];
char dir[100];
int main()
{
    memset(vst,0,sizeof(vst));
    dir['U']=0,dir['R']=1,dir['D']=2,dir['L']=3;
    int h,w;
    scanf("%d%d",&h,&w);
    int sx,sy;
    for(int i=0; i<h; i++)
    {
        char t[15];
        scanf("%s",t);
        for(int j=0; j<w; j++)
        {
            g[i][j]=t[j];
            if(t[j]=='U' || t[j]=='R' || t[j]=='D' || t[j]=='L')
                sx=i, sy=j;
        }
    }
    int d=dir[g[sx][sy]];

    while(1)
    {
        //if(vst[nx][ny][d]) break;
        if(vst[sx][sy][d]) break;
        vst[sx][sy][d]=true;
        int nx=sx+mx[d],ny=sy+my[d];
        if(nx<0 || nx>=h || ny<0 || ny>=w)
        {
            d=(d+1)%4;
            continue;
        }
        if(g[nx][ny]!='*')
        {
            d=(d+1)%4;
            //vst[sx][sy][d]=true;
            continue;
        }
        //vst[nx][ny][d]=true;
        sx=nx, sy=ny;
    }
    int ans=0;
    for(int i=0; i<h; i++)
    {
        for(int j=0; j<w; j++)
        {
```

```
        for(int k=0; k<4; k++)
            if(vst[i][j][k])
            {
                ans++;
                break;
            }
    }
    printf("%d\n",ans);
    return 0;
}
```