C题题解

题目

身为一名魔法少女，助人为乐显然是举手之劳。

魔法少女今天又遇到了一个问题。在她手上现在正拿着一个由 n\times m$n\times m$ 个正方形元件所组成的 nn 行 mm 列 的魔法元件。每个正方形元件初始都有着2条呈对角线形状的传送门，分别对应着正方形元件的两条对角线。初始状态下，每个元件只会激活一个传送门。

现在这个元件会将左上角连接到能量输出端，右下角连到能量输入端，从第 00 秒开始能量开始由输出端输出，魔法少女从第 11 秒开始每秒可以施展一次魔法，将其中一个元件的传送门改变为另一条对角线对应的传送门。

现在她想知道输入端最早能在什么时候得到能量，这个问题魔法少女显然一下子就秒了，但她想考验下你能否求出答案。

Standard Input

第 11 行两个数 n, m\ (1\le n, m \leq 500)n,m (1≤n,m≤500)

如题意所示在接下来的 nn 行中, 每行有 mm 个字符。每个字符均为 `\` 或 `/`, 表示魔法元件上魔法管道的连接方向。

Standard Output

输出一个数, 表示所需要的最短时间。

如果永远无法得到能量，请输出 `NO SOLUTION` (**注意输出格式**)

Samples

Input	Output
`3 5 \\/\\ \\/// /\\\\`	`1`

Note

请注意**时间限制**并尽量优化你的算法。

Problem ID	2711
Problem Title	魔法少女
Time Limit	200 ms
Memory Limit	128 MiB
Output Limit	64 MiB
Source	2022 UESTC ICPC Training for Graph

题解

把格子棋盘抽象为点阵，如果我们可以确定，对于第(i,j)个格子，存在两种情况：

- 若$mp[i,j]$是/则将点$i*(m+1)+j$和点$(i-1)*(m+1)+j+1$连0边
  将点$(i-1)*(m+1)+j$和点$i*(m+1)+j+1$连1边
- 相反的话反过来01值来连接。

然后我们用迪杰斯特拉跑最短路即可

~~注意判断no solution的情况~~

代码

```
#include <bits/stdc++.h>
using namespace std;
const int MAXN=1000005;
int dis[MAXN];
int head[MAXN],ver[MAXN],nxt[MAXN],wi[MAXN],tot;
inline void add(int x,int y,int z)
{ver[++tot]=y;nxt[tot]=head[x];head[x]=tot;wi[tot]=z;}
char mp[505][505];
int n,m;
inline void dj()
{
    priority_queue<pair<int,int>>q;
    dis[1]=0;
    q.push({0,1});
    while(q.size())
    {
        int x=q.top().second;int v=q.top().first;
        q.pop();
        // cout<<x<<endl;
        if(dis[x]<-v)continue;
        for(int i=head[x];i;i=nxt[i])
        {
            int y=ver[i];
            if(dis[y]>dis[x]+wi[i])
            {
                dis[y]=dis[x]+wi[i];
                q.push({-dis[y],y});
            }
        }
    }
}
int main()
{
    scanf("%d %d",&n,&m);
    for(int i=1;i<=n;i++)
        scanf("%s",mp[i]+1);
    memset(dis,0x3f3f3f,sizeof(dis));
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=m;j++)
        {
            if(mp[i][j]=='/')
            {
                add(i*(m+1)+j,(i-1)*(m+1)+j+1,0);
                add((i-1)*(m+1)+j+1,i*(m+1)+j,0);
```

```
                add((i-1)*(m+1)+j,i*(m+1)+j+1,1);
                add(i*(m+1)+j+1,(i-1)*(m+1)+j,1);
            }
            else{
                add(i*(m+1)+j,(i-1)*(m+1)+j+1,1);
                add((i-1)*(m+1)+j+1,i*(m+1)+j,1);
                add((i-1)*(m+1)+j,i*(m+1)+j+1,0);
                add(i*(m+1)+j+1,(i-1)*(m+1)+j,0);
            }
        }
    }
    dj();
    if(dis[(n+1)*(m+1)]>500000)cout<<"NO SOLUTION"<<endl;
    else cout<<dis[(n+1)*(m+1)]<<endl;
    return 0;
}
```