P1518 [USACO2.4]两只塔姆沃斯牛 The Tamworth Two

题目描述

两只牛逃跑到了森林里。Farmer John 开始用他的专家技术追捕这两头牛。你的任务是模拟他们的行为（牛和 John）。

追击在 10×10 的平面网格内进行。一个格子可以是：一个障碍物，两头牛（它们总在一起），或者 Farmer John。两头牛和 Farmer John 可以在同一个格子内（当他们相遇时），但是他们都不能进入有障碍的格子。

一个格子可以是：

. 空地；
* 障碍物；
C 两头牛；
F Farmer John。
这里有一个地图的例子：

```
*...*.....
......*...
...*...*..
..........
...*.F....
*.....*...
...*......
..C......*
...*.*....
.*.*......
```

牛在地图里以固定的方式游荡。每分钟，它们可以向前移动或是转弯。如果前方无障碍（地图边沿也是障碍），它们会按照原来的方向前进一步。否则它们会用这一分钟顺时针转 90 度。 同时，它们不会离开地图。

Farmer John 深知牛的移动方法，他也这么移动。

输入格式

输入共十行，每行 10 个字符，表示如上文描述的地图。

输出格式

输出一个数字，表示 John 需要多少时间才能抓住牛们。如果 John 无法抓住牛，则输出 0。

输入样例

....
......*...
...*...*..
....
...*.F....
....
...*......
..C......*
...*.*....
.*.*......

输出样例

49

本题也是一道基础模拟题，关键的难点在于循环的终止方法。农夫和奶牛都按照规则的方法进行行走，很有可能会进入某种循环路线而无法相遇，那么我们该如何判断呢？很明显，如果奶牛和者农夫按照某个方向走过某个重复的格子，那么我们就认为它们进入了循环。

基本的思路是制作一个标志位，但是这个标志位该如何定义呢？

这里一般提供两种普遍的方法：

1、将双方的坐标和角度按照一个固定的计算方法进行存储，例如：

农民x + 农民y*10 + 牛x * 100 + 牛y * 1000 + 农民朝向 * 10000 + 牛朝向 * 100000

这种方法需要开很大的空间

2、使用六维数组进行存储，这种比较简单

编码

```
#include<bits/stdc++.h>
using namespace std;
char m[12][12];//地图
//数组第0位存储方向
//数组第1,2位存储当前坐标
int f[3], c[3];
int ans; //秒数
bool zt[160005];//记录专属值是否出现

//移动函数
void move(int x, int y, int mi, int h) {
    if (mi == 0) {
        //遇到障碍更新方向
       if (m[x - 1][y] == '*') {
            if (h == 0) {
                f[0] = 1;
            } else {
                c[0] = 1;
            }
        }
            //向上移动
       else if (h == 0) {
            f[1]--;
        } else {
            c[1]--;
        }
    } else if (mi == 1) {
        if (m[x][y + 1] == '*') {
            if (h == 0) {
                f[0] = 2;
            } else {
                c[0] = 2;
            }
        } else if (h == 0) {
            f[2]++;
        } else {
            c[2]++;
        }
    } else if (mi == 2) {
        if (m[x + 1][y] == '*') {
            if (h == 0) {
                f[0] = 3;
            } else {
                c[0] = 3;
            }
        } else if (h == 0) {
            f[1]++;
```

```
        } else {
            c[1]++;
        }
    } else {
        if (m[x][y - 1] == '*') {
            if (h == 0) {
                f[0] = 0;
            } else {
                c[0] = 0;
            }
        } else if (h == 0) {
            f[2]--;
        } else {
            c[2]--;
        }
    }
}
//判断循环终止条件：如果奶牛坐标与农夫坐标相等，则他们重叠，返回0，退出循环
bool pd() {
    if (f[1] == c[1] && f[2] == c[2]) {
        return false;
    } else {
        return true;
    }
}
int main() {
    //在最外面包上一圈星号，直接通过星号判断越界问题
   for (int i = 0; i <= 11; i++) {
        m[i][0] = '*', m[i][11] = '*';
    }
    for (int i = 1; i <= 11; i++) {
        m[0][i] = '*', m[11][i] = '*';
    }
    for (int i = 1; i <= 10; i++) {
        for (int j = 1; j <= 10; j++) {
            cin >> m[i][j];
            //记录农夫和奶牛的初始坐标
          if (m[i][j] == 'F') {
                f[1] = i, f[2] = j;
            }
            if (m[i][j] == 'C') {
                c[1] = i, c[2] = j;
            }
        }
    }
    while (pd()) {//模拟每秒
       int tdz = f[1] + f[2] * 10 + c[1] * 100 + c[2] * 1000 + f[0]
            * 10000 + c[0] * 40000;
```

```
            //之前这个位置来过，说明进入了死循环
            if (zt[tdz]) {
                cout << 0 << endl;
                return 0;
            }
            zt[tdz] = 1;//标记
            //依次移动农夫和奶牛
            move(f[1], f[2], f[0], 0);
            move(c[1], c[2], c[0], 1);
            ans++;//记录秒数
        }
        cout << ans << endl;//输出
        return 0;
}
```