

sshwy

登出 (<http://www.zhengruioi.com/logout?>[_token=NKmndiWDTKcAqYw2zTdeJWMs8qqxXopmluSke6K4ectr4p7oIm0Nx9kfVPUa\)](#)

ZROI

好评 差评
[+4]

正睿2018暑期集训CD班附加赛Round3暨范里吉吉纪念赛

B. 吉吉的跳跃

描述

提交

自定义测试

统计 (</contest/78/problem/302/statistics>)[返回比赛 \(/contest/78\)](/contest/78)

题目背景

吉吉是一个很跳的人，今天他惹怒了某 C 姓男子。于是他不得不逃往教室旁边的小树林.....

题目描述

吉吉被传送到了一个森林中。森林是一个 $n \times m$ 的网格，每个格子可能是空地（用 $.$ 表示），也可能是障碍物（用 $\#$ 表示）。吉吉一直珍藏着森林的地图，他能经过空地，但是他不能经过障碍物，也不能跑出森林。他事先在森林中设置了很多关键点（用 $+$ 表示）。一开始吉吉会被传送到任意一个位置，如果他走到他设置的关键点，他就安全了。

因为吉吉是一个很跳的人，所以他的移动方式也十分奇怪。假设吉吉的初始位置是第 a 行第 b 列，那么他可以钦定两个正整数 x 和 y ，并将 $a \leftarrow a + x, b \leftarrow b + y$ 。其中 x, y 需要满足 $\min(x, y) = 0$ ，并且对于所有的 $a \leq i \leq a + x, b \leq j \leq b + y$ 都有第 i 行第 j 列为空地。也就是说，吉吉每次可以跳过一个线段，满足线段上的所有格子都是空地。

由于某 C 姓男子正在追捕他，所以吉吉想尽快回到他设置的关键点。你的任务就是对每个吉吉可能被传送到位置（即： $1 \leq i \leq n, 1 \leq j \leq m$ ），都算出他到最近一个关键点的最短距离。

输入格式

第一行两个正整数 n 和 m ，意义见题目描述。

接下来 n 行，每行一个长度为 m 的字符串，描述森林的地图。

输出格式

输出 n 行，每行 m 个整数或字符，表示吉吉如果在第 i 行第 j 列，他到最近一个关键点的最短距离。如果该点为障碍物，输出 #；如果从该点出发吉吉不可能跳到关键点，输出 x；如果该点为关键点，输出 0；否则输出吉吉到关键点的最短距离。

样例 1

input

```
10 10
+...#.#..
.#.###.###
...#...+..
##.#.#.#.#
+...#...#
.##.###.##
....#+...#
#..##..#..
####..#.#.
...#.#....
```

output

```
0 1 1 1 1 # 2 # X X
1 # 2 # # # 2 # # #
1 2 2 # 1 1 1 0 1 1
# # 2 # 2 # 2 # 2 #
0 # 2 3 2 # 2 2 2 #
1 # # 3 # # # 2 # #
1 2 2 2 # 0 1 1 1 #
# 3 3 # # 1 2 # 2 3
# # # # 2 1 # 6 # 4
X X X # 3 # 5 5 5 4
```

样例 2

input

```

20 20
+.#...#...#...#...#
#...#...#...#...#...+
+.#...#...#...#...#
#####
...#...#.....#...#
.+.....#...#...#...
...#.....#.....#
#####
..#...#.....#.....#
.+..#...#...#...#...#
...#.....#.....#...+
#####
+#.....#...#.#...#+
.#.###.#.#.###.#.##.
....#....#....#....
#####
+.....
.....
.....
.....
    
```

output

```

0 1 # 4 5 5 # 8 9 9 # 7 7 6 # 3 3 2 # 1
# 2 3 3 # 6 7 7 # 9 9 8 # 5 5 4 # 1 1 0
0 1 # 4 5 5 # 8 9 9 # 7 7 6 # 3 3 2 # 1
# # # # # # # # # # # # # # # # # # #
2 1 2 # 2 2 2 # 4 4 4 5 5 5 5 # 8 8 8 #
1 0 1 1 1 1 1 # 4 4 4 # 6 6 6 # 8 8 8 9
2 1 2 # 2 2 2 3 3 3 3 # 6 6 6 7 7 7 7 #
# # # # # # # # # # # # # # # # # # #
2 1 # 2 3 3 # 7 6 7 7 6 # 3 3 3 3 2 # X
1 0 1 1 # 4 5 # 6 7 # 6 6 # 4 4 # 2 2 #
2 1 2 # 5 4 5 5 5 # 5 5 5 5 4 # 1 1 1 0
# # # # # # # # # # # # # # # # # # #
0 # 3 4 4 4 4 # 7 8 7 # X # 4 4 3 4 # 0
1 # 3 # # # 5 # 7 # 7 # # # 5 # 3 # # 1
1 2 2 2 # 6 5 6 6 # 6 6 6 6 5 # 2 2 2 1
# # # # # # # # # # # # # # # # # # #
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
    
```

限制与规定

- 对于 10% 的数据, $n, m \leq 5$ 。
- 对于 30% 的数据, $n, m \leq 60$ 。
- 对于 60% 的数据, $n, m \leq 200$ 。
- 对于 100% 的数据, $n, m \leq 2000$ 。
- 数据有一定梯度。

时间限制: 3s

空间限制: 512MB

提示

由于输入输出量较大, 你可以使用给定的输入输出优化代码:

```

#include <bits/stdc++.h>
using namespace std;

const int OutputBufferSize = 10000000;
namespace input {
    #define BUF_SIZE 100000
    #define OUT_SIZE 100000
    #define ll long long
    bool IOerror = 0;
    inline char nc() {
        static char buf[BUF_SIZE], *p1 = buf + BUF_SIZE, *pend = buf + BUF_SIZE;
        if (p1 == pend) {
            p1 = buf; pend = buf + fread(buf, 1, BUF_SIZE, stdin);
            if (pend == p1) { IOerror = 1; return -1; }
        }
        return *p1++;
    }
    inline bool blank(char ch) {
        return ch == ' ' || ch == '\n' || ch == '\r' || ch == '\t';
    }
    inline void read(char &ch) {
        ch = nc();
        while (blank(ch)) ch = nc();
    }
    inline void read(int &x) {
        char ch = nc(); x = 0;
        for (; blank(ch); ch = nc());
        if (IOerror) return;
        for (; ch >= '0' && ch <= '9'; ch = nc()) x = x * 10 + ch - '0';
    }
    #undef ll
    #undef OUT_SIZE
    #undef BUF_SIZE
};

namespace output {
    char buffer[OutputBufferSize];
    char *s = buffer;
    inline void flush() {
        assert(stdout != NULL);
        fwrite(buffer, 1, s - buffer, stdout);
        s = buffer;
        fflush(stdout);
    }
    inline void print(const char ch) {
        if (s - buffer > OutputBufferSize - 2) flush();
        *s++ = ch;
    }
    inline void print(char* str) {
        while (*str != 0) print(char(*str++));
    }
    inline void print(int x) {
        char buf[25] = {0}, *p = buf;
        if (x == 0) print('0');
        while (x) *(++p) = x % 10, x /= 10;
        while (p != buf) print(char(*(p--) + '0'));
    }
}

```



```
}

const int maxn = 2005;
const int inf = 0x3f3f3f3f;
char s[maxn][maxn];
int n, m, a[maxn][maxn];

int main() {
    input::read(n), input::read(m);
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            input::read(s[i][j]);
        }
    }

    /*
    Your Program Here
    */

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            if (a[i][j] == -1) {
                output::print('#');
            } else if (a[i][j] == inf) {
                output::print('X');
            } else {
                output::print(a[i][j]);
            }
            output::print(" \n"[j == m]);
        }
    }
    output::flush();
    return 0;
}
```

 (<http://www.zhengrui.com/contest/78/problem/302?locale=zh-cn>)  (<http://www.zhengrui.com/contest/78/problem/302?locale=en>)

Zhengrui Online Judge

Server time: 2018-08-21 08:37:52 | 浙ICP备17047493号