Solved Programming Problems

অন্তহীন ছুটে চলা, অবিরাম জীবনের অর্থ খুজে ফেরা

APRIL 1, 2011 BY SHAHAB

UVa : 11624 (Fire!)

i
1 Vote

```
1  // http://uva.onlinejudge.org/external/116/11624.html (http:
2  // Runtime: .508s
3  // Tag: Bfs
4
5
6  /*
7   * File:   main.cpp
8   * Author: shahab
9   * Created on April 1, 2011, 3:43 PM
10  */
11
12 // @BEGIN_OF_SOURCE_CODE
13
14 #include <iostream>
15 #include <cstdio>
16 #include <algorithm>
17 #include <cstring>
18 #include <string>
19 #include <cctype>
20 #include <stack>
21 #include <queue>
22 #include <list>
23 #include <vector>
24 #include <map>
25 #include <sstream>
```

```cpp
#include <cmath>
#include <bitset>
#include <utility>
#include <set>
#include <numeric>

#define INF_MAX 2147483647
#define INF_MIN -2147483647
#define pi acos(-1.0)
#define N 1000000
#define LL long long

#define For(i, a, b) for ( int i = (a); i < (b); i++ )
#define Fors(i, sz) for ( size_t i = 0; i < sz.size (); i++
#define Set(a, s) memset (a, s, sizeof (a))

using namespace std;

struct state {
    int x;
    int y;
    int c;

    state (int p, int q, int r) {
        x = p;
        y = q;
        c = r;
    }

    state () { }
} a;

int row, col;
char matrix [1000 + 5] [1000 + 5];
int cost_joe [1000 + 5] [1000 + 5];
int cost_fire [1000 + 5] [1000 + 5];
int dr [] = {-1, 0, 1, 0};
int dc [] = {0, 1, 0, -1};

void bfs_for_joe (int r, int c)
{
    queue <state> q;
    cost_joe [r] [c] = 0;
    q.push(state (r, c, 0));

    while ( !q.empty() ) {
        a = q.front(); q.pop();

        for ( int i = 0; i < 4; i++ ) {
            int nx = a.x + dr [i];
            int ny = a.y + dc [i];
            if ( nx >= 0 && nx < row && ny >= 0 && ny < col
                cost_joe [nx] [ny] = a.c + 1;
                q.push(state (nx, ny, a.c + 1));
            }
        }
    }
```

```
 83  }
 84
 85  void bfs_for_fire ()
 86  {
 87      queue <state> q;
 88
 89      for ( int i = 0; i < row; i++ ) {
 90          for ( int j = 0; j < col; j++ ) {
 91              if ( matrix [i] [j] == 'F' ) {
 92                  cost_fire [i] [j] = 0;
 93                  q.push(state (i, j, 0));
 94              }
 95          }
 96      }
 97
 98      while ( !q.empty() ) {
 99          a = q.front(); q.pop();
100
101          for ( int i = 0; i < 4; i++ ) {
102              int nx = a.x + dr [i];
103              int ny = a.y + dc [i];
104              if ( nx >= 0 && nx < row && ny >= 0 && ny < col
105                  cost_fire [nx] [ny] = a.c + 1;
106                  q.push(state (nx, ny, a.c + 1));
107              }
108          }
109      }
110  }
111
112  void reset ()
113  {
114      for ( int i = 0; i < 1005; i++ ) {
115          for ( int j = 0; j < 1005; j++ )
116              cost_joe [i] [j] = cost_fire [i] [j] = INF_MAX;
117      }
118  }
119
120
121  int main(int argc, char** argv)
122  {
123      //freopen ("in.txt", "r", stdin);
124      //freopen ("out.txt", "w", stdout);
125
126      int testCase;
127      scanf ("%d", &testCase);
128
129      while ( testCase-- ) {
130          scanf ("%d %d", &row, &col);
131
132          for ( int i = 0; i < row; i++ ) scanf ("%s", matrix
133
134          reset ();
135
136          for ( int i = 0; i < row; i++ ) {
137              for ( int j = 0; j < col; j++ ) {
138                  if ( matrix [i] [j] == 'J' ) {
139                      bfs_for_joe (i, j);
```

```
140                         i = row;
141                         j = col;
142                     }
143                 }
144             }
145 
146             bfs_for_fire();
147 
148             int minimum_escape_time = INF_MAX;
149 
150             // 1st & last row
151             for ( int i = 0; i < col; i++ ) {
152                 if ( cost_fire [0] [i] > cost_joe [0] [i] ) mini
153                 if ( cost_fire [row - 1] [i] > cost_joe [row - 1
154             }
155 
156             // 1st & last col
157             for ( int i = 0; i < row; i++ ) {
158                 if ( cost_fire [i] [0] > cost_joe [i] [0] ) mini
159                 if ( cost_fire [i] [col - 1] > cost_joe [i] [col
160             }
161 
162             if ( minimum_escape_time == INF_MAX ) printf ("IMPOS
163             else printf ("%d\n", minimum_escape_time + 1);
164 
165         }
166 
167         return 0;
168     }
169 
170     // @END_OF_SOURCE_CODE
```