插头 DP 大字典

CMU_xiaodao

Last build at October 22, 2015

Contents

-1.1 多条回路

主要两大模型：路径问题，和染色问题。先来看各种路径模型。这类问题实际上不需要保存状态的连通性信息，只需要用 01 状态记录轮廓线上插头是否存在，因而严格意义上只能算是轮廓线 DP。不过因为和其他路径问题紧密相关，可以拿来参照学习，里面的技巧也出来先了其他问题中。

-1.1.1 HDU 1693. Eat the Trees

入门题。

-1.1.2 ZJU 4231. The Hive II

格子变成了六边形，建议竖着做。

-1.2 一条回路

最经典的模型，可以比较一下括号表示，和最小表示，因为最小表示解决的问题更多，转移写起来也更简单。

-1.2.1 Ural 1519. Formula 1

-1.2.2 HDU 1964. Pipes

-1.2.3 FZU 1977. Pandora adventure

很多题输入的时候会对格子进行区分，比如必须经过的点、可以经过的点、和障碍。比如这个题。

```
1
2  //}/* ..... */
3
4  const int N = int(10), M = 1<<18, _M = 3;
5  char A[N+1][N+1]; int n, m;
6  int b[N+1], bb[N+1];
7
8  LL encode(){
9      FLC(bb, -1); int n = 1; bb[0] = 0; LL s = 0;
10     DWN(i, m+1, 0){
11         if (!~bb[b[i]]) bb[b[i]] = n++;
12         b[i] = bb[b[i]];
13         s <<= _M; s |= b[i];
14     }
15     return s;
16 }
17 void decode(LL s){
18     REP(i, m+1){
19         b[i] = s & _U(_M);
20         s >>= _M;
21     }
22 }
23
24
25
26 const int Prime = 9979, MaxSize = M; LL d;
27 struct hashTable{
28     LL state[MaxSize]; int key[MaxSize];
29     int hd[Prime], nxt[MaxSize], sz;
30     void clear(){
31         sz = 0;
32         FLC(hd, -1);
33     }
```

```
34      void add(){
35          LL s = encode();
36          int x = s % Prime;
37
38          for (int i=hd[x];~i;i=nxt[i]){
39              if (state[i] == s){
40                  INC(key[i], d);
41                  return;
42              }
43          }
44
45          state[sz] = s, key[sz] = d;
46          nxt[sz] = hd[x], hd[x] = sz;
47          ++sz;
48
49          return;
50      }
51      void roll(){
52          REP(ii, sz) state[ii] <<= _M;
53      }
54
55  } H[2]; int src, des;
56
57
58  bool AA[N+1][N+1]; int Dist[N][N], Ways[N][N];
59
60  int Qx[N*N], Qy[N*N], cz, op;
61  int step, step0, ways;
62
63
64  void gao(int sx, int sy){
65
66
67      src = 0, des = 1; H[des].clear();
68      RST(b); d = 1; H[des].add();
69
70      REP(i, n){
71          REP(j, m){
72              if (!AA[i][j]) continue;
73
74              swap(src, des); H[des].clear();
75
76
77              bool dn = AA[i+1][j];
78              bool rt = AA[i][j+1];
79              bool in = (A[i][j] == 'O' || i == sx && j == sy);
80
81
82              //cout << dn << " " << rt << " " << in << endl;
83              //cout << endl;
84
85              REP(ii, H[src].sz){
86                  decode(H[src].state[ii]);
87                  d = H[src].key[ii];
88
89                  int lt = b[j], up = b[j+1];
90
91                  // cout << lt << " " << up << endl;
92
93
94                  if (lt && up){
95                      if (lt == up){
96
97                      }
98                      else{
99                          if (!in){
100                             REP(k, m+1) if (b[k] == lt) b[k] = up;
```

```
101                             b[j] = b[j+1] = 0;
102                             H[des].add();
103                         }
104                     }
105                 }
106                 else if (lt || up){
107
108                     if (in){
109                         b[j] = b[j+1] = 0; H[des].add();
110                     }
111                     else{
112                         int t = lt | up;
113                         if (dn){
114                             b[j] = t; b[j+1] = 0; H[des].add();
115                         }
116                         if (rt){
117                             b[j] = 0; b[j+1] = t; H[des].add();
118                         }
119                     }
120                 }
121                 else{
122
123                     if (in){
124                         if (dn){
125                             b[j] = m; b[j+1] = 0;
126                             H[des].add();
127                         }
128                         if (rt){
129                             b[j] = 0; b[j+1] = m;
130                             H[des].add();
131                         }
132                     }
133                     else{
134                         if (dn && rt){
135                             b[j] = b[j+1] = m;
136                             H[des].add();
137                         }
138                     }
139
140
141                 }
142             }
143         }
144         H[des].roll();
145     }
146
147 }
148
149 void solve(){
150     step = INF, step0 = 0, ways = 0;
151
152     RD(n, m); RST(AA); cz = 0, op = 1;
153     RST(Dist, Ways); REP_2(i, j, n, m){
154         RC(A[i][j]); if (A[i][j] == 'X'){
155             Qx[0] = i, Qy[0] = j;
156             Dist[i][j] = 1;
157             Ways[i][j] = 1;
158         }
159         AA[i][j] = A[i][j] == '#' || A[i][j] == 'O';
160         if (A[i][j] == '#') step0 += 1;
161     }
162
163
164     while (cz < op){
165
166         int ux = Qx[cz], uy = Qy[cz], du = Dist[ux][uy]; ++cz;
167         if (du > step) break;
```

```
168
169         if (A[ux][uy] == '#'){
170             gao(ux, uy);
171             assert(H[des].sz <= 1);
172             if (H[des].sz == 1){
173                 step = du;
174                 INC(ways, pdt(Ways[ux][uy], H[des].key[0]));
175             }
176             continue;
177         }
178
179         if (A[ux][uy] == 'O'){
180             if (!step0){
181                 step = du, ways = Ways[ux][uy];
182                 break;
183             }
184             continue;
185         }
186
187         REP(i, 4){
188             int vx = ux + dx[i], vy = uy + dy[i];
189             if (0 <= vx && vx < n && 0 <= vy && vy < m){
190                 if (Dist[vx][vy] == du + 1) INC(Ways[vx][vy], Ways[ux][uy]);
191                 else if (!Dist[vx][vy]){
192                     Dist[vx][vy] = du + 1;
193                     Ways[vx][vy] = Ways[ux][uy];
194                     Qx[op] = vx;
195                     Qy[op] = vy;
196                     ++op;
197
198                 }
199             }
200         }
201     }
202
203     /* cout << endl;
204      REP_2(i, j, n, m){
205      cout << Ways[i][j] << " ";
206      if (j == m-1) cout << endl;
207      }
208      */
209
210     if (step == INF) printf("%d\n", -1);
211     else printf("%d %d\n", step + step0 - 1, ways);
212 }
213
214 int main(){
215
216 #ifndef ONLINE_JUDGE
217     freopen("/users/xiaodao/desktop/Exercise/in.txt", "r", stdin);
218     //freopen("out.txt", "w", stdout);
219 #endif
220     Rush{
221         printf("Case #%d: ", ++Case);
222         solve();
223     }
224 }
```

-1.2.4 第九届北航程序设计大赛现场决赛 - 晴天小猪当导游

再上题的基础上，每个格子所能继承和发出的插头也要考虑。

```
1
2
3 //}/* ..... */
4
```

```
5  const int N = 12, M = 1 << 20, _M = 3;
6
7  int n, m;
8  int b[N+1], bb[N+1];
9
10 LL encode(){
11     FLC(bb, -1); int n = 1; bb[0] = 0; LL s = 0;
12     DWN(i, m+1, 0){
13         if (!~bb[b[i]]) bb[b[i]] = n++;
14         b[i] = bb[b[i]];
15         s <<= _M; s |= b[i];
16     }
17     return s;
18 }
19
20 void decode(LL s){
21     REP(i, m+1){
22         b[i] = s & _U(_M);
23         s >>= _M;
24     }
25 }
26
27 const int Prime = 131071;
28
29 int d; struct hashTable{
30     LL state[M]; int key[M]; int sz;
31     int hd[Prime], nxt[M];
32
33     void clear(){
34         sz = 0;
35         FLC(hd, -1);
36     }
37
38     void push(){
39         LL s = encode();
40         int x = s % Prime;
41
42         for (int i=hd[x];~i;i=nxt[i]){
43             if (state[i] == s){
44                 INC(key[i], d);
45                 return;
46             }
47         }
48         state[sz] = s; key[sz] = d;
49         nxt[sz] = hd[x]; hd[x] = sz;
50         ++sz;
51         assert(sz < M);
52     }
53
54     void roll(){
55         REP(ii, sz) state[ii] <<= _M;
56     }
57 } H[2]; int src, des;
58 int A[N+1][N+1], tx, ty;
59
60
61 bool isBlock(int s){return s & 1;}
62 bool isMust(int s){return s & 2;}
63 bool isRight(int s){return s & 4;}
64 bool isLeft(int s){return s & 8;}
65 bool isDown(int s){return s & 16;}
66 bool isUp(int s){return s & 32;}
67
68 void init(){
69     tx = -1, ty = -1; RD(n, m); REP_2(i, j, n, m) if (isMust(RD(A[i][j]))) tx = i, ty = j;
70     REP(i, m) A[n][i] = 1; REP(i, n) A[i][m] = 1;
71 }
```

```
72
73  void solve(){
74      src = 0, des = 1; H[des].clear(); RST(b); d = 1; H[des].push();
75
76      int z = 0; REP(i, n){
77          REP(j, m){
78
79              if (isBlock(A[i][j])) continue;
80              swap(src, des); H[des].clear();
81
82              //cout << i << " " << j << " "<< H[src].sz << endl;
83
84              REP(ii, H[src].sz){
85                  decode(H[src].state[ii]); d = H[src].key[ii];
86
87                  int lt = b[j], up = b[j+1];
88                  bool dn = isDown(A[i][j]) && isUp(A[i+1][j]) && !isBlock(A[i+1][j]);
89                  bool rt = isRight(A[i][j]) && isLeft(A[i][j+1]) && !isBlock(A[i][j+1]);
90
91                  if (lt && up){
92                      if (lt == up){
93                          if (i*m+j>=tx*m+ty){
94                              int cnt = 0; REP(jj, m+1) if (b[jj]) ++cnt; // ?
95                              if (cnt == 2) INC(z, d);
96                          }
97                      }
98                      else{
99                          b[j] = b[j+1] = 0;
100                         REP(jj, m+1) if (b[jj] == lt) b[jj] = up;
101                         H[des].push();
102                     }
103                 }
104                 else if (lt || up){
105                     int t = lt | up;
106                     if (dn){
107                         b[j] = t; b[j+1] = 0;
108                         H[des].push();
109                     }
110                     if (rt){
111                         b[j] = 0; b[j+1] = t;
112                         H[des].push();
113                     }
114                 }
115                 else{
116
117                     if (!isMust(A[i][j])){
118                         H[des].push();
119                     }
120
121                     if (dn && rt){
122                         b[j] = b[j+1] = N-1;
123                         H[des].push();
124                     }
125                 }
126             }
127         }
128         H[des].roll();
129     }
130
131     printf("%d\n", z);
132 }
133
134
135 int main(){
136
137
138
```

```
    Rush{
        init();
        solve();
    }
}
```

-1.3 简单路径

不再是回路，需要多加状态记录独立插头生成的数目，而且插头不再总是成对出现。

对于加维，如果只是加比较简单的状态，可以开在 Hash 表下标里，比较方便。复杂的话就和联通信息一起放在 Hash 表里，可以多开数组，也可以一起压缩到状态里，后者 Roll() 的时候要分开。

-1.3.1 ZJU 3213. Beautiful Meadow

```
#include <bits/stdc++.h>
using namespace std;
#define REP(i, n) for (int i=0;i<n;++i)
#define RST(A) memset(A, 0, sizeof(A))
#define FLC(A, x) memset(A, x, sizeof(A))
typedef long long LL;
const int INF = 0x3f3f3f3f;
template<class T> void checkMax(T& a, T b){if (b > a) a = b;}

const int M = 10, _M = 3, _UM = 7;
int A[M+1][M+1];
int n, m;

int b[M+1], bb[M+1];
LL encode(){
    FLC(bb, -1); int n = 1; REP(i, n) bb[i] = 0; LL s = 0;
    for (int i=m+1;i>=0;--i){
        if (!~bb[b[i]]) bb[b[i]] = n++;
        s <<= _M; s |= bb[b[i]];
    }
    return s;
}
void decode(LL s){
    REP(i, m+1){
        b[i] = s & _UM; s >>= _M;
    }
}

const int Prime = 9979, MaxSize = 1 << 18;
int d;
struct hashTable{
    LL state[MaxSize]; int key[MaxSize];
    int hd[Prime], nxt[MaxSize], sz;
    void clear(){
        sz = 0;
        FLC(hd, -1);
    }
    void push(){
        LL s = encode();
        int x = s % Prime;

        for (int i=hd[x];~i;i=nxt[i]) if (state[i] == s){
            checkMax(key[i], d);
            return;
        }
        state[sz] = s; key[sz] = d;
        nxt[sz] = hd[x]; hd[x] = sz;
        ++sz;
        assert(sz < MaxSize);
    }
```

```
51      void roll(){
52          REP(ii, sz) state[ii] <<= _M;
53      }
54  } H[2][3]; int src, des;
55
56  void push(int c){
57      H[des][c].push();
58  }
59
60  int solve(){
61      scanf("%d %d", &n, &m); FLC(A, 0x80);
62      int z = 0; REP(i, n) REP(j, m){
63          scanf("%d", &A[i][j]);
64          if (!A[i][j]) A[i][j] = -INF;
65          checkMax(z, A[i][j]);
66      }
67
68      src = 0, des = 1; REP(c, 3) H[des][c].clear();
69      RST(b); d = 0; H[des][0].push();
70
71      REP(i, n){
72          REP(j, m){
73              if (A[i][j] <= 0) continue; // skip
74              swap(src, des); REP(c, 3) H[des][c].clear();
75              REP(c, 3) REP(ii, H[src][c].sz){
76                  decode(H[src][c].state[ii]); d = H[src][c].key[ii] + A[i][j];
77                  int lt = b[j], up = b[j+1];
78                  bool dn = A[i+1][j] >= 0, rt = A[i][j+1] >= 0;
79                  if (lt && up){
80                      if (lt == up){
81
82                      }
83                      else{
84                          REP(i, m+1) if (b[i] == lt) b[i] = up;
85                          b[j] = b[j+1] = 0;
86                          push(c);
87                      }
88                  }
89                  else if (lt || up){
90                      int t = lt | up;
91                      if (dn){
92                          b[j] = t; b[j+1] = 0;
93                          push(c);
94                      }
95                      if (rt){
96                          b[j] = 0; b[j+1] = t;
97                          push(c);
98                      }
99                      if (c != 2){ // 独立插头——消失
100                         b[j] = b[j+1] = 0;
101                         push(c+1);
102                     }
103                 }
104                 else{
105                     d -= A[i][j]; H[des][c].push(); d += A[i][j]; // skip
106                     if (dn && rt){
107                         b[j] = b[j+1] = m;
108                         push(c);
109                     }
110                     if (c < 2){ // 独立插头——生成
111                         if (dn){
112                             b[j] = m; b[j+1] = 0;
113                             push(c+1);
114                         }
115                         if (rt){
116                             b[j] = 0; b[j+1] = m;
117                             push(c+1);
```

```
                        }
                    }
                }
            }
        }
        REP(c, 3) H[des][c].roll();
    }

    assert(H[des][2].sz <= 1);
    REP(ii, H[des][2].sz) checkMax(z, H[des][2].key[ii]);
    return z;
}

int main(){
    int T; cin >> T; while (T--){
        cout << solve() << endl;
    }
}
```

-1.3.2 POJ 1739. Tony's Tour

楼教的男人八题之一，根据题意算是简单路径，但是确定了起始点和终点，本质上是一条回路问题，不需要记录独立插头的数目。（可以在起点和终点时修改成独立插头的转移。而不必加一圈外围构造）

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include <cassert>
using namespace std;
#define REP(i, n) for (int i=0;i<n;++i)
#define RST(A) memset(A, 0, sizeof(A))
#define FLC(A, x) memset(A, x, sizeof(A))
typedef long long LL;
const int INF = 0x3f3f3f3f;
template<class T> void checkMax(T& a, T b){if (b > a) a = b;}

const int M = 10, _M = 3, _UM = 7;
int A[M+1][M+1];
int n, m;

int b[M+1], bb[M+1];
LL encode(){
    FLC(bb, -1); int n = 1; REP(i, n) bb[i] = 0; LL s = 0;
    for (int i=m+1;i>=0;--i){
        if (!~bb[b[i]]) bb[b[i]] = n++;
        s <<= _M; s |= bb[b[i]];
    }
    return s;
}
void decode(LL s){
    REP(i, m+1){
        b[i] = s & _UM; s >>= _M;
    }
}

const int Prime = 9979, MaxSize = 1 << 18;
int d;
struct hashTable{
    LL state[MaxSize]; int key[MaxSize];
    int hd[Prime], nxt[MaxSize], sz;
    void clear(){
        sz = 0;
        FLC(hd, -1);
    }
    void push(){
        LL s = encode();
```

```
43          int x = s % Prime;
44
45          for (int i=hd[x];~i;i=nxt[i]) if (state[i] == s){
46              key[i] += d;
47              return;
48          }
49          state[sz] = s; key[sz] = d;
50          nxt[sz] = hd[x]; hd[x] = sz;
51          ++sz;
52          assert(sz < MaxSize);
53      }
54      void roll(){
55          REP(ii, sz) state[ii] <<= _M;
56      }
57  } H[2]; int src, des; int tx, ty;
58
59  void push(){
60      H[des].push();
61  }
62
63  int solve(){
64      RST(A); int z = 0; REP(i, n) REP(j, m){
65          char c; scanf(" %c", &c);
66          A[n-i-1][j] = c == '.';
67      }
68
69      REP(i, n) REP(j, m) if (A[i][j]){
70          tx = i, ty = j;
71      }
72
73      src = 0, des = 1; H[des].clear();
74      RST(b); b[1] = b[m] = 1; d = 1; H[des].push();
75
76      REP(i, n){
77          REP(j, m){
78              if (!A[i][j]) continue; // skip
79              swap(src, des); H[des].clear();
80              REP(ii, H[src].sz){
81                  decode(H[src].state[ii]); d = H[src].key[ii];
82                  int lt = b[j], up = b[j+1];
83                  bool dn = A[i+1][j], rt = A[i][j+1];
84                  if (lt && up){
85                      if (lt == up){
86                          if (i == tx && j == ty){
87                              z += d;
88                          }
89                      }
90                      else{
91                          REP(i, m+1) if (b[i] == lt) b[i] = up;
92                          b[j] = b[j+1] = 0;
93                          push();
94                      }
95                  }
96                  else if (lt || up){
97                      int t = lt | up;
98                      if (dn){
99                          b[j] = t; b[j+1] = 0;
100                         push();
101                     }
102                     if (rt){
103                         b[j] = 0; b[j+1] = t;
104                         push();
105                     }
106                 }
107                 else{
108                     if (dn && rt){
109                         b[j] = b[j+1] = m;
```

```
                        push();
                    }
                }
            }
        }
        H[des].roll();
    }

    assert(H[des].sz <= 1);
    return z;
}

int main(){

    //freopen("in.txt", "r", stdin);

    while (~scanf("%d %d", &n, &m) && n){
        cout << solve() << endl;
    }
}
```

-1.3.3 POJ 3133. Manhattan Wiring

```
//}/* ..... */

const int N = int(9), M = 1 << (N+5), _M = 3;

int A[N+1][N+1]; int n, m;
int b[N+1], bb[N+1];

LL encode(){
    FLC(bb, -1); int n = 3; REP(i, n) bb[i] = i; LL s = 0;
    DWN(i, m+1, 0){
        if (!~bb[b[i]]) bb[b[i]] = n++;
        b[i] = bb[b[i]];
        s <<= _M; s |= b[i];
    }
    return s;
}

void decode(LL s){
    REP(i, m+1){
        b[i] = s & _U(_M);
        s >>= _M;
    }
}

const int Prime = 9979, MaxSize = M; int d;
struct hashTable{
    LL state[MaxSize]; int key[MaxSize];
    int hd[Prime], nxt[MaxSize], sz;
    void clear(){
        sz = 0;
        FLC(hd, -1);
    }
    int add(){
        LL s = encode();
        int x = s % Prime;

        for (int i=hd[x];~i;i=nxt[i]){
            if (state[i] == s){
                checkMin(key[i], d);
                return key[i];
            }
        }
```

```
43
44          state[sz] = s, key[sz] = d;
45          nxt[sz] = hd[x], hd[x] = sz;
46          ++sz;
47          assert(sz < MaxSize);
48          return key[sz-1];
49      }
50  } H[2]; int src, des;
51
52  int solve(){
53
54      FLC(A, -1); REP_2(i, j, n, m){
55          RD(A[i][j]); if (A[i][j] == 1) A[i][j] = -1;
56          else if (A[i][j] > 1) --A[i][j];
57      }
58      // -1 障碍
59      // 1, 2: 特殊插头
60
61      src = 0, des = 1; H[des].clear(); RST(b); d = 0; H[des].add();
62      REP(i, n){
63
64          REP(j, m){
65
66              if (!~A[i][j]) continue;
67              swap(src, des); H[des].clear();
68
69              REP(ii, H[src].sz){
70
71                  decode(H[src].state[ii]); d = H[src].key[ii] + 1;
72                  int lt = b[j], up = b[j+1];
73
74                  if (lt && up){
75
76                      if (A[i][j]) continue;
77
78                      if (lt == up){
79                          if (lt <= 2){
80                              b[j] = b[j+1] = 0;
81                              H[des].add();
82                          }
83                      }
84                      else{
85
86                          if (lt <= 2){
87                              if (up <= 2) continue; // 冲突...
88                              swap(lt, up); // 合并插头，优先取独立插头
89                          }
90                          REP(k, m+1) if (b[k] == lt) b[k] = up;
91                          b[j] = b[j+1] = 0;
92                          H[des].add();
93                      }
94                  }
95                  else if (lt || up){
96
97                      int t = lt | up;
98
99                      if (A[i][j]){
100                         if (t <= 2){
101                             if (t == A[i][j]){
102                                 b[j] = b[j+1] = 0; H[des].add();
103                             }
104                         }
105                         else{
106                             REP(k, m+1) if (b[k] == t) b[k] = A[i][j];
107                             b[j] = b[j+1] = 0; H[des].add();
108                         }
109
```

```
                    }
                    else{
                        if (~A[i+1][j]){
                            b[j] = t; b[j+1] = 0; H[des].add();
                        }
                        if (~A[i][j+1]){
                            b[j] = 0; b[j+1] = t; H[des].add();
                        }
                    }
                }
                else{

                    if (A[i][j]){
                        if (~A[i+1][j]){
                            b[j] = A[i][j]; b[j+1] = 0;
                            H[des].add();
                        }

                        if (~A[i][j+1]){
                            b[j] = 0; b[j+1] = A[i][j];
                            H[des].add();
                        }
                    }
                    else{

                        --d; H[des].add(); ++d;

                        if (~A[i+1][j] && ~A[i][j+1]){
                            b[j] = b[j+1] = max(3, m);
                            H[des].add();
                        }
                    }
                }
            }
        }

        REP(ii, H[des].sz) H[des].state[ii] <<= _M;
    }

    RST(b); d = INF; int z = H[des].add();
    if (z == INF) z = 0; else z -= 2;
    return z;
}



int main(){

#ifndef ONLINE_JUDGE
    freopen("/users/xiaodao/desktop/Exercise/in.txt", "r", stdin);
    //freopen("/users/xiaodao/desktop/Exercise/out.txt", "w", stdout);
#endif

    while (~scanf("%d%d", &n, &m) && n){
        OT(solve());
        //break;
    }
}
```

-1.4 染色模型

-1.4.1 Topcoder SRM 312. Div1 CheapestIsland

```

const int N = 12, M = 1 << 18, _M = 3;
```

```
3
4  int n, m;
5  int b[N+1], bb[N+1];
6  void decode(LL s){
7      REP(i, m+1){
8          b[i] = s & _U(_M); s >>= _M;
9      }
10 }
11 LL encode(){
12     FLC(bb, -1); int n = 1; bb[0] = 0; LL s = 0;
13     DWN(i, m+1, 0){
14         if (!~bb[b[i]]) bb[b[i]] = n++;
15         s <<= _M; s |= bb[b[i]];
16     }
17     return s;
18 }
19 const int Prime = 9979;
20 int A[N+1][N+1];
21 int i, j; LL d; int op; struct hashMap{
22     LL state[M]; LL key[M]; int sz;
23     int hd[Prime]; int nxt[M];
24     void clear(){
25         sz = 0;
26         FLC(hd, -1);
27     }
28     void push(){
29         LL s = encode();
30         int x = s % Prime;
31         LL d = ::d; if(op) d += A[i][j];
32         for (int i=hd[x];~i;i=nxt[i]){
33             if (state[i] == s){
34                 checkMin(key[i], d);
35                 return;
36             }
37         }
38         state[sz] = s; key[sz] = d;
39         nxt[sz] = hd[x]; hd[x] = sz;
40         ++sz;
41         assert(sz < M);
42     }
43     void roll(){
44         LL U = _U(_M*(m+1));
45         REP(ii, sz){
46             state[ii] <<= _M;
47             state[ii] &= U;
48         }
49     }
50     void display(int ii){
51         decode(state[ii]);
52         cout << key[ii] << ": " << endl;
53         REP(i, m+1) cout << b[i] << " "; cout << endl;
54         puts("");
55     }
56 } H[2]; int src, des;
57
58 LL z; int ii; void trans(){
59     decode(H[src].state[ii]); d = H[src].key[ii];
60     int lt = j ? b[j-1] : 0, lu = b[j], up = b[j+1];
61
62     if (!op){
63         b[j] = 0;
64         if (up){
65             int c1 = 0, c2 = 0; REP(i, m+1){
66                 if (b[i]) ++c1;
67                 if (b[i] == up) ++c2;
68             }
69             if (c2 == 1){
```

```
                    if (c1 == 1) checkMin(z, d);
                    return;
                }
            }
            H[des].push();
        }
        else{
            if (lt && up){
                if (lt != up){
                    REP(i, m+1) if (b[i] == up) b[i] = lt;
                }
                b[j] = lt;
                H[des].push();
            }
            else if (lt || up){
                int t = lt | up;
                b[j] = t;
                H[des].push();
            }
            else{
                b[j] = m;
                H[des].push();
            }
        }
    }

    int solve(){
        src = 0; des = 1; H[des].clear(); RST(b); d = 0; op = 0; H[des].push();
        z = 0; REP_N(i, n+1){
            REP_N(j, m){
                swap(src, des); H[des].clear();
                //cout << i << " " << j << ": " << H[src].sz << endl;
                REP_N(ii, H[src].sz){
                    //H[src].display(ii);
                    op = 0; trans();
                    op = 1; trans();
                }
            }
            H[des].roll();
        }

        return int(z);
    }

    class CheapestIsland {
    public:
        int minCost(vector <string> cells) {

            RST(A); n = int(cells.size()); REP(i, n){
                m = 0; istringstream iss(cells[i]);
                while (iss >> A[i][m]) ++m;
            }
            return solve();
        }
    };
```

-1.4.2 UVA 10572. black & white

```

    //}/* ..... */


    // ?
    const int N = 8, M = 1 << (20), _Mc = 2, _Mb = 4;

```

```
8  int A[N][N];
9  int n, m;
10
11 int c[N+2];
12 int b[N+2], bb[N+3];
13
14 LL encode(){
15     FLC(bb, -1); int n = 1; bb[0] = 0; LL s = 0;
16     DWN(i, m+1, 0){
17         if (!~bb[b[i]]) bb[b[i]] = n++;
18         b[i] = bb[b[i]];
19         s <<= _Mb; s |= b[i];
20     }
21     DWN(i, m+1, 0){
22         s <<= _Mc; s |= c[i];
23     }
24     return s;
25 }
26
27 void decode(LL s){
28     REP(i, m+1){
29         c[i] = s & _U(_Mc);
30         s >>= _Mc;
31     }
32     REP(i, m+1){
33         b[i] = s & _U(_Mb);
34         s >>= _Mb;
35     }
36 }
37
38 const int Prime = 9979, MaxSize = M;
39
40 LL sta[N*N+9][MaxSize];
41 int pre[N*N+9][MaxSize];
42
43 LL d; int u; int i, j; struct hashMap{
44
45     LL state[MaxSize], key[MaxSize]; int sz;
46     int hd[Prime], nxt[MaxSize];
47
48     void clear(){
49         sz = 0;
50         FLC(hd, -1);
51     }
52
53     void push(){
54
55         LL s = encode();
56         int x = s % Prime;
57
58         for (int i=hd[x];~i;i=nxt[i]){
59             if (state[i] == s){
60                 key[i] += d;
61                 return;
62             }
63         }
64         state[sz] = s; key[sz] = d;
65         nxt[sz] = hd[x], hd[x] = sz;
66
67         sta[i*m+j][sz] = s;
68         pre[i*m+j][sz] = u;
69
70         ++sz;
71         assert(sz < MaxSize);
72         return;
73     }
74
```

```
75      void roll(){
76
77          LL Uc = _U(_Mc*(m+1)), Ub = _U(_Mb*(m+1)) << (_Mc*(m+1));
78
79          REP(ii, sz){
80              LL s = state[ii], sc = s & Uc, sb = s & Ub;
81              sc <<= _Mc; sc &= Uc; sb <<= _Mb; sb &= Ub;
82              state[ii] = sc | sb;
83          }
84      }
85
86      void display(){
87          cout << sz << ": ";
88          cout << endl;
89          REP(ii, sz){
90              cout << state[ii] << " " << key[ii] << endl;
91              decode(state[ii]);
92              REP(i, m+1) cout << c[i] << " "; cout << endl;
93              REP(i, m+1) cout << b[i] << " "; cout << endl;
94          }
95          cout << endl;
96      }
97
98
99  } H[2]; int src, des;
100
101
102 int cc(char c){
103     if (c == '#') return 1;
104     if (c == 'o') return 2;
105     return 0;
106 }
107
108 bool legal(int cc){
109
110     if (cc == c[j+1]) return true;
111     //if (i == 0) return true;
112     int up = b[j+1]; if (!up) return true;
113     int c1 = 0, c2 = 0;
114
115     REP(i, m+1) if (i != j+1){
116         if (b[i] == b[j+1]){
117             assert(c[i] == c[j+1]);
118         }
119         if (c[i] == c[j+1] && b[i] == b[j+1]) ++c1;
120         if (c[i] == c[j+1]) ++c2;
121     }
122
123     if (!c1){
124         if (c2) return false;
125         if (i < n-1 || j < m-2) return false;
126     }
127     return true;
128 }
129
130 void trans(int ii, int cc){
131
132     LL s = H[src].state[ii]; d = H[src].key[ii]; u = ii; decode(s);
133
134     int lf = j ? c[j-1] : 0, lu = c[j], up = c[j+1];
135     c[j] = cc;
136
137     if (lf == cc && up == cc){
138         if (lu == cc) return;
139         int lf_b = b[j-1], up_b = b[j+1];
140         REP(i, m+1) if (b[i] == lf_b || b[i] == up_b){
141             b[i] = lf_b;
```

```
142             }
143             b[j] = lf_b;
144         }
145         else if (lf == cc || up == cc){
146             if (lf == cc) b[j] = b[j-1]; else b[j] = b[j+1];
147         }
148         else{
149             if (i == n-1 && j == m-1 && lu == cc) return;
150             b[j] = m+2;
151         }
152 
153         if (!legal(cc)) return;
154         H[des].push();
155         return;
156     }
157 
158     char Board[N+1][N+1];
159 
160     void print(int u){
161         RST(Board); DWN(i, n*m, 0){
162             decode(sta[i][u]);
163             Board[i/m][i%m] = (c[i%m] == 1 ? '#' : 'o');
164             u = pre[i][u];
165         }
166         REP(i, n) puts(Board[i]);
167     }
168 
169 
170     void solve(){
171 
172         RD(n, m); RST(A); REP_2(i, j, n, m) A[i][j] = cc(RC());
173         src = 0, des = 1; H[des].clear(); RST(b); RST(c); d = 1; H[des].push();
174 
175         REP_N(i, n){
176             REP_N(j, m){
177 
178                 swap(src, des); H[des].clear();
179 
180                 // cout << " " << i << " " << j << ": " << endl;
181                 // H[src].display();
182 
183                 REP(ii, H[src].sz){
184 
185                     if (!A[i][j]){
186                         trans(ii, 1);
187                         trans(ii, 2);
188                     }
189                     else if (A[i][j] == 1){
190                         trans(ii, 1);
191                     }
192                     else if (A[i][j] == 2){
193                         trans(ii, 2);
194                     }
195                 }
196             }
197 
198             H[des].roll();
199         }
200 
201         //H[des].display();
202 
203         LL z = 0; int t; REP(ii, H[des].sz){
204             decode(H[des].state[ii]);
205 
206             //int cnt = 0; RST(bb); REP_1(i, m) if (!bb[b[i]]) bb[b[i]] = 1, ++cnt;
207             int cnt = 0; REP(i, m+1) if (b[i] > cnt) checkMax(cnt, b[i]);
208 
```

```
209             if (cnt <= 2){
210                 z += H[des].key[ii];
211                 t = ii;
212             }
213         }
214
215         OT(z); if (z) print(t);
216         puts("");
217         return;
218     }
219
220
221     int main(){
222
223     #ifndef ONLINE_JUDGE
224         freopen("/users/xiaodao/desktop/Exercise/in.txt", "r", stdin);
225         //freopen("/users/xiaodao/desktop/Exercise/out.txt", "w", stdout);
226     #endif
227
228         Rush{
229             solve();
230             //break;
231         }
232     }
```

-1.5 综合问题

-1.5.1 ZJU 3256. Tour in the Castle

-1.5.2 BZOJ 1494: [NOI2007] 生成树计数

```
1
2   const int N = 6, _M = 3;
3
4   int m; LL n;
5   int b[N+1], bb[N+1];
6
7   int encode(){
8       FLC(bb, -1); int n = 1; REP(i, n) bb[i] = i; int s = 0;
9       DWN_1(i, m, 1){
10          if (!~bb[b[i]]) bb[b[i]] = n++;
11          s <<= _M; s |= bb[b[i]];
12      }
13      return s;
14  }
15
16  void decode(int s){
17      REP(i, m){
18          b[i] = s & _U(_M); s >>= _M;
19      }
20  }
21
22  bool ck(){
23      if (b[0]){
24          bool ok = false;
25          REP_1(i, m) if (b[i] == b[0]){
26              ok = true;
27              break;
28          }
29          return ok;
30      }
31      return true;
```

```
32  }
33
34  const int Prime = 9979, MaxSize = 1 << 18; int d;
35  struct hashTable{
36      int state[MaxSize]; int key[MaxSize];
37      int hd[Prime], nxt[MaxSize], sz;
38
39      void clear(){
40          sz = 0;
41          FLC(hd, -1);
42      }
43
44      void push(){
45
46          if (!ck()) return;
47          int s = encode();
48          int x = s % Prime;
49          for (int i=hd[x];~i;i=nxt[i]) if (state[i] == s){
50              INC(key[i], d);
51              return;
52          }
53          state[sz] = s; key[sz] = d;
54          nxt[sz] = hd[x]; hd[x] = sz;
55          ++sz;
56          assert(sz < MaxSize);
57          return;
58      }
59      void roll(){
60          REP(ii, sz) state[ii] >>= _M;
61      }
62      void display(int ii){
63          decode(state[ii]);
64          REP(i, m) cout << b[i] << " ";
65          cout << ": " << key[ii];
66          cout << endl;
67
68      }
69  } H[2]; int src, des;
70
71  void push(){
72      H[des].push();
73  }
74  void clear(){
75      H[des].clear();
76  }
77  void roll(){
78      H[des].roll();
79  }
80
81  int ii; void decode(){
82      decode(H[src].state[ii]); d = H[src].key[ii];
83  }
84
85
86
87
88  int Hid[1<<18]; MT::matrix A, B;
89  bool flag; // dfs for matrix build?
90
91  void dfs(int i = 0){
92      if (i == m){
93          if (flag){
94              if (ck()) ++B.d[ii][Hid[encode()]];
95          }
96          else push();
97      }
98      else{
```

```
99          dfs(i+1);
100         if (b[m] == m+1){
101             if (b[i]){
102                 b[m] = b[i];
103                 dfs(i+1);
104                 b[m] = m+1;
105             }
106         }
107         else{
108             if (b[i] && b[i] != b[m]){
109                 VI bb; bb.resize(m);
110                 REP(j, m) bb[j] = b[j];
111                 REP(j, m) if (b[j] == bb[i]) b[j] = b[m];
112                 dfs(i+1);
113                 REP(j, m) b[j] = bb[j];
114             }
115         }
116     }
117 }
118
119 void init(){
120
121     src = 0, des = 1; clear();
122     RST(b); b[m] = 1; d = 1; push();
123
124     flag = 0;
125
126     FOR(i, 1, m){
127         swap(src, des); clear();
128         REP_N(ii, H[src].sz){
129             //H[src].display(ii);
130             decode(); b[m] = m+1; dfs();
131         }
132     }
133
134     swap(src, des); MT::n = H[src].sz; A.init(); REP_N(ii, H[src].sz){
135         //H[src].display(ii);
136         Hid[H[src].state[ii]] = ii;
137         A.d[0][ii] = H[src].key[ii];
138     }
139
140     flag = 1; B.init(); REP_N(ii, H[src].sz){
141         decode(); b[m] = m+1; dfs();
142     }
143 }
144
145 int solve(){
146     init(); A *= pow(B, n-m);
147     REP_1(i, m) b[i] = 1;
148     return A.d[0][Hid[encode()]];
149 }
150
151
152 int main(){
153
154 #ifndef ONLINE_JUDGE
155     freopen("/users/xiaodao/desktop/Exercise/in.txt", "r", stdin);
156     //freopen("out.txt", "w", stdout);
157 #endif
158
159     MOD = 65521;
160
161     while (~scanf("%d %lld", &m, &n)){
162         printf("%d\n", solve());
163     }
164
165 }
```

-1.5.3 Pipeline Plans

-1.5.4 UVA 10531. Maze Statistics

```
1
2
3  //}/* ..... */
4
5  const int N = int(10), _M = 3;
6  char A[N+1][N+1]; int n, m;
7  int b[N+1], bb[N+1];
8
9  LL encode(){
10     FLC(bb, -1); int n = 2; REP(i, n) bb[i] = i; LL s = 0;
11     DWN(i, m+1, 0){
12         if (!~bb[b[i]]) bb[b[i]] = n++;
13         b[i] = bb[b[i]];
14         s <<= _M; s |= b[i];
15     }
16     return s;
17 }
18 void decode(LL s){
19     REP(i, m+1){
20         b[i] = s & _U(_M);
21         s >>= _M;
22     }
23 }
24
25 const int Prime = 9979, MaxSize = 1 << 18; DB d;
26 struct hashTable{
27     LL state[MaxSize]; DB key[MaxSize];
28     int hd[Prime], nxt[MaxSize], sz;
29     void clear(){
30         sz = 0;
31         FLC(hd, -1);
32     }
33     void add(){
34         LL s = encode();
35         int i; REP_N(i, m+1) if (b[i] == 1) break;
36         if (i == m+1) return;
37
38         int x = s % Prime;
39
40         for (int i=hd[x];~i;i=nxt[i]){
41             if (state[i] == s){
42                 key[i] += d;
43                 return;
44             }
45         }
46
47         state[sz] = s, key[sz] = d;
48         nxt[sz] = hd[x], hd[x] = sz;
49         ++sz;
50         assert(sz < MaxSize);
51         return;
52     }
53     void roll(){
54         //REP(ii, sz) state[ii] <<= _M;
55     }
56     void display(int ii){
57         decode(state[ii]);
```

```
58          REP(i, m+1) cout << b[i] << " ";
59          cout << ": " << key[ii] << endl;
60          puts("");
61      }
62  } H[2]; int src, des;
63
64  DB isBarrier[N][N], ans[N][N];
65  int i, j, ii;
66
67  void putBarrier(){
68      decode(H[src].state[ii]); d = H[src].key[ii] * isBarrier[i][j]; if (!sgn(d)) return;
69      b[j+1] = 0; H[des].add();
70
71  }
72  void putBlank(){
73      decode(H[src].state[ii]); d = H[src].key[ii] * (1 - isBarrier[i][j]); if (!sgn(d)) return;
74      int lt = b[j], up = b[j+1];
75      if (lt && up){
76          if (lt != up){
77              if (lt < up) swap(lt, up);
78              REP(i, m+1) if (b[i] == lt) b[i] = up;
79          }
80      }
81      else if (lt || up){
82          b[j+1] = lt | up;
83      }
84      else{
85          b[j+1] = m+1;
86      }
87      H[des].add();
88  }
89
90  DB connectedProb(){
91
92      src = 0, des = 1; H[des].clear();
93      RST(b); b[1] = 1; d = 1; H[des].add();
94
95      REP_N(i, n){
96          REP_N(j, m){
97              swap(src, des); H[des].clear();
98              REP_N(ii, H[src].sz){
99                  putBarrier();
100                 putBlank();
101             }
102         }
103     }
104
105     DB z = 0; REP(ii, H[des].sz){
106         decode(H[des].state[ii]);
107         if (b[m] == 1) z += H[des].key[ii];
108     }
109     return z;
110 }
111
112 void solve(){
113
114     RD(n, m); REP_2(i, j, n, m) RF(isBarrier[i][j]);
115
116     DB total = connectedProb(); assert(sgn(total)>0);
117
118     REP(i, n){
119         REP(j, m){
120             DB cache = isBarrier[i][j];
121             isBarrier[i][j] = 1.0;
122             printf("%.6f%c", connectedProb()*cache / total, j == m-1 ? '\n' : ' ');
123             isBarrier[i][j] = cache;
124         }
```

```
125     }
126 }
127
128 int main(){
129
130 #ifndef ONLINE_JUDGE
131     freopen("in.txt", "r", stdin);
132     //freopen("out.txt", "w", stdout);
133 #endif
134     Rush{
135         if (Case++) puts("");
136         solve();
137     }
138 }
```

-1.5.5 FZU 2199. Patchmania I

```
1
2
3  //}/* ..... */
4
5  const int N = int(10), M = 1<<18, _M = 3;
6  char A[N+1][N+1]; int n, m;
7  int b[N+1], bb[N+1];
8
9  LL encode(){
10     FLC(bb, -1); int n = 1; bb[0] = 0; LL s = 0;
11     DWN(i, m+1, 0){
12         if (!~bb[b[i]]) bb[b[i]] = n++;
13         b[i] = bb[b[i]];
14         s <<= _M; s |= b[i];
15     }
16     return s;
17 }
18 void decode(LL s){
19     REP(i, m+1){
20         b[i] = s & _U(_M);
21         s >>= _M;
22     }
23 }
24
25
26
27 const int Prime = 9979, MaxSize = M; LL d;
28 struct hashTable{
29     LL state[MaxSize]; int key[MaxSize];
30     int hd[Prime], nxt[MaxSize], sz;
31     void clear(){
32         sz = 0;
33         FLC(hd, -1);
34     }
35     void add(){
36         LL s = encode();
37         int x = s % Prime;
38
39         for (int i=hd[x];~i;i=nxt[i]){
40             if (state[i] == s){
41                 INC(key[i], d);
42                 return;
43             }
44         }
45
46         state[sz] = s, key[sz] = d;
47         nxt[sz] = hd[x], hd[x] = sz;
48         ++sz;
```

```
49
50          return;
51      }
52      void roll(){
53          REP(ii, sz) state[ii] <<= _M;
54      }
55
56  } H[2]; int src, des;
57
58
59  bool AA[N+1][N+1]; int Dist[N][N], Ways[N][N];
60
61  int Qx[N*N], Qy[N*N], cz, op;
62  int step, step0, ways;
63
64
65  void gao(int sx, int sy){
66
67
68      src = 0, des = 1; H[des].clear();
69      RST(b); d = 1; H[des].add();
70
71      REP(i, n){
72          REP(j, m){
73              if (!AA[i][j]) continue;
74
75              swap(src, des); H[des].clear();
76
77
78              bool dn = AA[i+1][j];
79              bool rt = AA[i][j+1];
80              bool in = (A[i][j] == 'O' || i == sx && j == sy);
81
82
83              //cout << dn << " " << rt << " " << in << endl;
84              //cout << endl;
85
86              REP(ii, H[src].sz){
87                  decode(H[src].state[ii]);
88                  d = H[src].key[ii];
89
90                  int lt = b[j], up = b[j+1];
91
92                  // cout << lt << " " << up << endl;
93
94
95                  if (lt && up){
96                      if (lt == up){
97
98                      }
99                      else{
100                         if (!in){
101                             REP(k, m+1) if (b[k] == lt) b[k] = up;
102                             b[j] = b[j+1] = 0;
103                             H[des].add();
104                         }
105                     }
106                 }
107                 else if (lt || up){
108
109                     if (in){
110                         b[j] = b[j+1] = 0; H[des].add();
111                     }
112                     else{
113                         int t = lt | up;
114                         if (dn){
115                             b[j] = t; b[j+1] = 0; H[des].add();
```

```
116                         }
117                         if (rt){
118                             b[j] = 0; b[j+1] = t; H[des].add();
119                         }
120                     }
121                 }
122                 else{
123 
124                     if (in){
125                         if (dn){
126                             b[j] = m; b[j+1] = 0;
127                             H[des].add();
128                         }
129                         if (rt){
130                             b[j] = 0; b[j+1] = m;
131                             H[des].add();
132                         }
133                     }
134                     else{
135                         if (dn && rt){
136                             b[j] = b[j+1] = m;
137                             H[des].add();
138                         }
139                     }
140 
141 
142                 }
143             }
144         }
145         H[des].roll();
146     }
147 
148 }
149 
150 void solve(){
151     step = INF, step0 = 0, ways = 0;
152 
153     RD(n, m); RST(AA); cz = 0, op = 1;
154     RST(Dist, Ways); REP_2(i, j, n, m){
155         RC(A[i][j]); if (A[i][j] == 'X'){
156             Qx[0] = i, Qy[0] = j;
157             Dist[i][j] = 1;
158             Ways[i][j] = 1;
159         }
160         AA[i][j] = A[i][j] == '#' || A[i][j] == 'O';
161         if (A[i][j] == '#') step0 += 1;
162     }
163 
164 
165     while (cz < op){
166 
167         int ux = Qx[cz], uy = Qy[cz], du = Dist[ux][uy]; ++cz;
168         if (du > step) break;
169 
170         if (A[ux][uy] == '#'){
171             gao(ux, uy);
172             assert(H[des].sz <= 1);
173             if (H[des].sz == 1){
174                 step = du;
175                 INC(ways, pdt(Ways[ux][uy], H[des].key[0]));
176             }
177             continue;
178         }
179 
180         if (A[ux][uy] == 'O'){
181             if (!step0){
182                 step = du, ways = Ways[ux][uy];
```

```
                    break;
                }
                continue;
            }

            REP(i, 4){
                int vx = ux + dx[i], vy = uy + dy[i];
                if (0 <= vx && vx < n && 0 <= vy && vy < m){
                    if (Dist[vx][vy] == du + 1) INC(Ways[vx][vy], Ways[ux][uy]);
                    else if (!Dist[vx][vy]){
                        Dist[vx][vy] = du + 1;
                        Ways[vx][vy] = Ways[ux][uy];
                        Qx[op] = vx;
                        Qy[op] = vy;
                        ++op;

                    }
                }
            }
        }

        /* cout << endl;
         REP_2(i, j, n, m){
         cout << Ways[i][j] << " ";
         if (j == m-1) cout << endl;
         }
         */

        if (step == INF) printf("%d\n", -1);
        else printf("%d %d\n", step + step0 - 1, ways);
    }

    int main(){

    #ifndef ONLINE_JUDGE
        freopen("/users/xiaodao/desktop/Exercise/in.txt", "r", stdin);
        //freopen("out.txt", "w", stdout);
    #endif
        Rush{
            printf("Case #%d: ", ++Case);
            solve();
        }
    }
```

-1.5.6 SDOI 2014. 电路板

```

    //}/* ..... */

    const int N = int(11), M = 1 << 18, _M = 6;

    int A[N+1][N+1]; int n, m, k;
    int b[N+1], bb[N+N+1];

    LL encode(){
        FLC(bb, -1); int n = k+1; REP(i, n) bb[i] = i; LL s = 0;
        DWN(i, m+1, 0){
            if (!~bb[b[i]]) bb[b[i]] = n++;
            b[i] = bb[b[i]];
            s <<= _M; s |= b[i];
        }
        return s;
    }

    void decode(LL s){
```

```
20      REP(i, m+1){
21          b[i] = s & _U(_M);
22          s >>= _M;
23      }
24  }
25
26  const int Prime = 9979, MaxSize = M;
27  VI Link[N][N];
28  int i, j; int dkey, dtot; struct hashTable{
29      LL state[MaxSize]; int key[MaxSize], tot[MaxSize];
30      int hd[Prime], nxt[MaxSize], sz;
31      void clear(){
32          sz = 0;
33          FLC(hd, -1);
34      }
35      void add(){
36
37
38          /*if (Link[i][j].size()){
39           if (b[j] && find(ALL(Link[i][j]), b[j]) == Link[i][j].end()) return;
40           if (b[j+1] && find(ALL(Link[i][j]), b[j+1]) == Link[i][j].end()) return;
41           }*/
42
43          /*if (!Link[i][j].empty()){
44           cout << "!!!!" << endl;
45           if (b[j] > k || b[j+1] > k) return;
46           }*/
47
48          // REP(i, m+1) cout << b[i] << " "; cout << endl;
49
50          int d = dkey + bool(b[j]) + bool(b[j+1]);
51          LL s = encode();
52          int x = s % Prime;
53
54          for (int i=hd[x];~i;i=nxt[i]){
55              if (state[i] == s){
56                  if (d < key[i]){
57                      key[i] = d;
58                      tot[i] = dtot;
59                  }
60                  else if (d == key[i]){
61                      INC(tot[i], dtot);
62                  }
63                  //return key[i];
64                  return;
65              }
66          }
67
68          state[sz] = s, key[sz] = d, tot[sz] = dtot;
69          nxt[sz] = hd[x], hd[x] = sz;
70          ++sz;
71          assert(sz < MaxSize);
72          //return key[sz-1];
73      }
74
75      void display(int ii){
76          decode(state[ii]);
77          cout << state[ii] << " "<< key[ii] << " " << tot[ii] << ": ";
78          REP(i, m+1) cout << b[i] << " "; cout <<endl;
79          cout << endl;
80      }
81
82      void roll(){
83          REP(ii, sz) state[ii] <<= _M;
84      }
85
86  } H[2]; int src, des;
```

```
87
88
89  bool bad;
90  void init(){
91      RST(A); REP_2(i, j, n, m) A[i][j] = 1 - RD(), Link[i][j].clear();
92      bad = false;
93      REP_1(i, k){
94          int a, b, c, d;
95          RD(a, b); Link[a][b].PB(i);
96          RD(c, d); Link[c][d].PB(i);
97          if (a == c && b == d) bad = true;
98      }
99      // cout << bad << endl;
100 }
101
102
103 int ii; int lt, up; bool dn, rt;
104
105 bool match(int t){
106     if (!t || t > k) return true;
107     return find(ALL(Link[i][j]), t) != Link[i][j].end();
108 }
109
110
111
112 void match(){
113     VI t; ECH(it, Link[i][j]) if (*it != lt && *it != up) t.PB(*it);
114
115     // cout << t.size() << " " << lt << " " << up << endl;
116
117
118     int _lt = lt, _up = up;
119     VI bb; REP(i, m+1) bb.PB(b[i]);
120
121     if (lt && lt > k){
122         ECH(it, t){
123             REP(i, m+1) if (b[i] == lt) b[i] = *it;
124             lt = *it; match();
125             REP(i, m+1) b[i] = bb[i]; lt = _lt; up = _up;
126         }
127         return;
128     }
129
130     if (up && up > k){
131         ECH(it, t){
132             REP(i, m+1) if (b[i] == up) b[i] = *it;
133             up = *it; match();
134             REP(i, m+1) b[i] = bb[i]; lt = _lt; up = _up;
135         }
136         return;
137     }
138
139
140     // cout << " " << t.size() << " " << endl;
141
142
143     if (t.size() > 2) return;
144     if (t.size() == 2){
145         if (!dn || !rt) return;
146         b[j] = t[0], b[j+1] = t[1];
147         H[des].add();
148         b[j] = t[1], b[j+1] = t[0];
149         H[des].add();
150     }
151     else if (t.size() == 1){
152
153         // cout << " " << dn << " " << rt << " " << t[0] << endl;
```

```
154
155         if (dn){
156             b[j] = t[0], b[j+1] = 0;
157             H[des].add();
158         }
159         if (rt){
160             b[j] = 0, b[j+1] = t[0];
161             H[des].add();
162         }
163     }
164     else if (t.size() == 0){
165
166         // if (i == 2 && j == 1) return;
167
168         // cout << " !!!!!: " << dtot << endl;
169
170         b[j] = b[j+1] = 0;
171         H[des].add();
172     }
173 }
174
175
176
177
178
179
180 void solve(){
181
182     if (bad){
183         cout << -1 << " " << 0 << endl;
184         return;
185     }
186
187     // 1 ' œ∞
188     // 0 Õ®—
189
190     src = 0, des = 1; H[des].clear(); RST(b); dkey = 0; dtot = 1; H[des].add();
191
192     REP_N(i, n){
193
194         REP_N(j, m){
195
196             if (!A[i][j]) continue;
197
198
199             swap(src, des); H[des].clear();
200
201
202             // cout << i << " "<< j << ": " << H[src].sz << endl;
203
204             REP_N(ii, H[src].sz){
205
206                 // H[src].display(ii);
207                 dkey = H[src].key[ii]; dtot = H[src].tot[ii];
208                 dn = A[i+1][j], rt = A[i][j+1];
209
210                 decode(H[src].state[ii]); lt = b[j], up = b[j+1];
211
212                 if (Link[i][j].size()){
213
214
215                     if (lt && up && lt == up) continue;
216                     // cout << " " << lt << " " << up << endl;
217
218
219                     if (!match(lt)) continue;
220                     if (!match(up)) continue;
```

```
                //cout << "!" << endl;
                // if (i == 2 && j == 1) continue;

                match();
            }
            else{



                if (lt && up){


                    if (lt == up){

                        // cout << " " << lt << " " << up << endl;
                        if (lt <= k){ //#
                            b[j] = b[j+1] = 0;
                            H[des].add();
                            if (dn && rt){

                                // cout << "!" << endl;

                                b[j] = b[j+1] = k+m;
                                H[des].add();
                            }
                        }
                    }
                    else{

                        if (dn && rt){
                            //b[j] = lt; b[j+1] = up;
                            H[des].add();
                            //b[j] = up; b[j+1] = lt; // Cross
                            //H[des].add();
                        }

                        if (lt <= k){
                            if (up <= k) continue;
                            swap(lt, up);
                        }

                        REP(jj, m+1) if (b[jj] == lt) b[jj] = up;
                        b[j] = b[j+1] = 0;
                        H[des].add();
                        if (dn && rt){
                            b[j] = b[j+1] = k+m;
                            H[des].add();
                        }
                    }
                }
                else if (lt || up){
                    int t = lt | up;
                    if (dn){
                        b[j] = t, b[j+1] = 0;
                        H[des].add();
                    }
                    if (rt){
                        b[j] = 0, b[j+1] = t;
                        H[des].add();
                    }
                }
                else{

                    b[j] = b[j+1] = 0;
                    H[des].add();

                    if (dn && rt){
```

```
288                             // cout << "??" << endl;
289                             b[j] = b[j+1] = k+m;
290                             H[des].add();
291                         }
292                     }
293                 }
294             }
295
296         }
297
298         H[des].roll();
299     }
300
301     if (H[des].sz == 0){
302         printf("-1 0\n");
303     }
304     else{
305         assert(H[des].sz == 1);
306         // H[des].display(0);
307         cout << H[des].key[0] << " " << H[des].tot[0] << endl;
308     }
309 }
310
311 int main(){
312
313
314     while (~scanf("%d%d%d", &n, &m, &k) && n){
315         init(); solve();
316         //break;
317     }
318 }
```

-1.6 神题

下面是一些个人感觉非常 tasty 的题目，他们要么是简洁但又艰深的问题，要么做完以后能给予我很大启发。

-1.6.1 HDU 4113. Construct the Great Wall

用两种方法都可以做，揭示了路径模型和染色模型之间的联系。

```
1  #include <iostream>
2  #include <cstdio>
3  #include <cstring>
4  #include <cassert>
5  using namespace std;
6  #define REP(i, n) for (int i=0;i<n;++i)
7  #define RST(A) memset(A, 0, sizeof(A))
8  #define FLC(A, x) memset(A, x, sizeof(A))
9  typedef long long LL;
10 const int INF = 0x3f3f3f3f;
11 template<class T> void checkMin(T& a, T b){if (b < a) a = b;}
12 template<class T> void checkMax(T& a, T b){if (b > a) a = b;}
13
14 const int M = 12, _M = 3, _UM = 7;
15 char A[M+1][M+1];
16 int n, m;
17
18 int b[M+1], bb[M+1];
19 LL encode(){
20     FLC(bb, -1); int n = 1; REP(i, n) bb[i] = 0; LL s = 0;
21     for (int i=m+1;i>=0;--i){
22         if (!~bb[b[i]]) bb[b[i]] = n++;
23         s <<= _M; s |= bb[b[i]];
24     }
25     return s;
```

```
26  }
27  void decode(LL s){
28      REP(i, m+1){
29          b[i] = s & _UM; s >>= _M;
30      }
31  }
32
33  const int Prime = 9979, MaxSize = 1 << 18;
34  int d;
35  struct hashTable{
36      LL state[MaxSize]; int key[MaxSize];
37      int hd[Prime], nxt[MaxSize], sz;
38      void clear(){
39          sz = 0;
40          FLC(hd, -1);
41      }
42      void push(){
43          LL s = encode();
44          int x = s % Prime;
45
46          for (int i=hd[x];~i;i=nxt[i]) if (state[i] == s){
47              checkMin(key[i], d);
48              return;
49          }
50          state[sz] = s; key[sz] = d;
51          nxt[sz] = hd[x]; hd[x] = sz;
52          ++sz;
53          assert(sz < MaxSize);
54      }
55      void roll(){
56          REP(ii, sz) state[ii] <<= _M;
57      }
58  } H[2][2]; int src, des; int tx, ty;
59
60  char Aij; void push(int c){
61      if (Aij == 'o' && !c) return;
62      if (Aij == 'x' && c) return;
63      H[des][c].push();
64  }
65
66  int solve(){
67      cin >> n >> m;
68      RST(A); int z = 0; REP(i, n) REP(j, m){
69          scanf(" %c", &A[i][j]);
70          if (A[i][j] == 'o') tx = i, ty = j;
71      }
72      ++n, ++m, ++tx, ++ty;
73
74      src = 0, des = 1; REP(c, 2) H[des][c].clear();
75      RST(b); d = 0; Aij = '.'; push(0); z = INF;
76
77      REP(i, n){
78          REP(j, m){
79              Aij = A[i][j]; swap(src, des); REP(c, 2) H[des][c].clear();
80              REP(c, 2) REP(ii, H[src][c].sz){
81                  decode(H[src][c].state[ii]); d = H[src][c].key[ii] + 1;
82                  int lt = b[j], up = b[j+1];
83                  bool dn = i != n-1, rt = j != m-1;
84                  if (lt && up){
85                      if (lt == up){
86                          int cnt = 0; REP(i, m+1) if (b[i]) ++cnt;
87                          if (cnt == 2 && i*m+j>=tx*m+ty){
88                              checkMin(z, d);
89                          }
90                      }
91                      else{
92                          b[j] = b[j+1] = 0; REP(i, m+1) if (b[i] == lt) b[i] = up;
```

```
93                          push(c);
94                      }
95                  }
96                  else if (lt || up){
97                      int t = lt | up;
98                      if (dn){
99                          b[j] = t; b[j+1] = 0;
100                         push(c^1);
101                     }
102                     if (rt){
103                         b[j] = 0; b[j+1] = t;
104                         push(c);
105                     }
106                 }
107                 else{
108                     --d; push(c); ++d;
109
110                     if (dn && rt){
111                         b[j] = b[j+1] = m;
112                         push(c^1);
113                     }
114                 }
115             }
116         }
117         REP(c, 2) H[des][c].roll();
118     }
119
120     REP(c, 2) assert(H[des][c].sz <= 1);
121     if (z == INF) z = -1;
122     return z;
123 }
124
125 int main(){
126
127 #ifndef ONLINE_JUDGE
128     freopen("in.txt", "r", stdin);
129     //freopen("out.txt", "w", stdout);
130 #endif
131     int T; cin >> T; REP(cas, T){
132         printf("Case #%d: ", cas+1);
133         cout << solve() << endl;
134     }
135 }
```

-1.6.2 ZOJ 2125/2156. Rocket Mania

```
1
2  //}/* ..... */
3
4
5  // ?
6  const int N = 9, M = 1 << (20), _M = 4;
7
8  int A[N+1][N+1]; int n, m;
9  int b[N+2], bb[N+2]; int BurNing;
10
11 LL encode(){
12     FLC(bb, -1); int n = 1; bb[0] = 0; LL s = BurNing;
13     DWN(i, m+1, 0){
14         if (!~bb[b[i]]) bb[b[i]] = n++;
15         b[i] = bb[b[i]];
16         s <<= _M; s |= b[i];
17     }
18     return s;
19 }
```

```
20
21  void decode(LL s){
22      REP(i, m+1){
23          b[i] = s & _U(_M);
24          s >>= _M;
25      }
26      BurNing = s;
27  }
28
29  const int Prime = 9979, MaxSize = M;
30  //#include <unordered_set>
31  //unordered_set<int> H[2];
32
33
34
35  struct hashMap{
36      LL state[MaxSize]; int sz;
37      int hd[Prime], nxt[MaxSize];
38      void clear(){
39          sz = 0;
40          FLC(hd, -1);
41      }
42      void push(LL s){
43          int x = s % Prime;
44
45          for (int i=hd[x];~i;i=nxt[i]){
46              if (state[i] == s) return;
47          }
48          state[sz] = s;
49          nxt[sz] = hd[x], hd[x] = sz;
50          ++sz;
51          assert(sz < MaxSize);
52          return;
53      }
54      void roll(){
55          REP(i, sz){
56
57              decode(state[i]);
58              assert(b[m] == 0);
59              assert(!_1(BurNing, m));
60
61              LL plug = state[i] & _U(m*_M);
62              state[i] ^= plug;
63              state[i] <<= 1; plug <<= _M;
64              state[i] |= plug;
65
66              //state[i] &= _U((m+1)*_M + (m+1));
67          }
68      }
69  } H[2];
70
71  int src, des;
72  vector<int> Pattern[5];
73
74  int cc(char c){
75      if (c == '-') return 1;
76      if (c == 'L') return 2;
77      if (c == 'T') return 3;
78      if (c == '+') return 4;
79      return 0;
80  }
81
82  int x;
83
84  void debug_state(){
85      REP(i, m+1) cout << b[i] << " "; cout << endl;
86      REP(i, m+1) if (_1(BurNing, i)) cout << '*' << " ";
```

```
87      else cout << '-' << " "; cout << endl;
88  }
89
90
91
92  //map<LL, LL> amazing;
93
94  int solve(){
95
96      n = 6; m = 9;
97      RST(A); REP_2(j, i, m, n) A[i][j] = cc(RC());
98      src = 0, des = 1; H[des].clear();
99      RST(b); b[x] = 1; BurNing |= _1(x);
100     H[des].push(encode());
101
102     REP(i, n){
103         REP(j, m){
104
105             swap(src, des); H[des].clear();
106
107             //cout << i << " " << j << ": " << H[src].sz << endl;
108
109             int best = -1; LL critical = 0;
110             vector<LL> _des;
111
112             REP(ii, H[src].sz){
113
114                 decode(H[src].state[ii]);
115                 //debug_state();
116                 //cout << "----" << endl;
117
118                 ECH(it, Pattern[A[i][j]]){
119
120                     decode(H[src].state[ii]);
121
122                     int p = *it;
123
124                     int lt = b[j], up = b[j+1];
125                     bool lt2 = _1(p,0) && lt, up2 = _1(p,1) && up;
126                     bool BurN = ((lt2 && _1(BurNing, j)) || (up2 && _1(BurNing, j+1)));
127
128                     b[j] = b[j+1] = 0;
129                     BurNing &= ~_1(j);
130                     BurNing &= ~_1(j+1);
131
132                     int t;
133
134                     if (lt2 && up2){
135                         t = up;
136                         REP(k, m+1) if (b[k] == lt || b[k] == up){
137                             if (BurN) BurNing |= _1(k);
138                             b[k] = t;
139                         }
140                     }
141                     else if (lt2 | up2){
142                         if (lt2) t = lt; else t = up;
143                     }
144                     else{
145                         t = m+1;
146                     }
147
148                     if (_1(p,2)){
149                         b[j] = t;
150                         if (BurN) BurNing |= _1(j);
151                     }
152
153                     if (_1(p,3) && j != m-1){
```

```
154                         b[j+1] = t;
155                         if (BurN) BurNing |= _1(j+1);
156                     }
157
158                     if (!BurNing) continue; // Cut-1
159
160                     RST(bb); REP(k, m+1) if (b[k]) ++bb[b[k]];
161                     REP(k, m+1) if (bb[b[k]] == 1 && !_1(BurNing, k)) b[k] = 0; // Cut-2
162
163                     LL s = encode(); _des.push_back(s);
164                     if (count_bits(BurNing) > best){
165                         best = count_bits(BurNing);
166                         critical = s;
167                     }
168                 }
169             }
170
171             decode(critical); int cBurNing = BurNing, cPlug = 0;
172             REP(k, m+1) if (b[k]) cPlug |= _1(k);
173
174             if (!_des.empty()){
175                 H[des].push(critical);
176
177                 ECH(it, _des){
178                     decode(*it); int Plug = 0;
179                     REP(k, m+1) if (b[k]) Plug |= _1(k);
180                     if ( (BurNing & cBurNing) == BurNing && (Plug & cPlug) == Plug){
181                         continue;
182                     }
183                     H[des].push(*it);
184                 }
185             }
186         }
187         H[des].roll();
188     }
189
190     int z = 0; REP(ii, H[des].sz){
191
192         checkMax(z, count_bits(H[des].state[ii]>>((m+1)*_M)));
193
194        /* if (count_bits(H[des].state[ii]>>((m+1)*_M)) == 7){
195             decode(H[des].state[ii]);
196             debug_state();
197         }*/
198     }
199     return z;
200 }
201
202 void Init(){
203
204     // 1 Left
205     // 2 Up
206     // 4 Down
207     // 8 Right
208
209     // '.'
210     Pattern[0].PB(0);
211     // '-'
212     Pattern[1].PB(1|8); Pattern[1].PB(2|4);
213     // 'L'
214     Pattern[2].PB(2|8); Pattern[2].PB(8|4); Pattern[2].PB(4|1); Pattern[2].PB(1|2);
215     // 'T'
216     int U = 15;
217     Pattern[3].PB(U^1); Pattern[3].PB(U^2); Pattern[3].PB(U^4); Pattern[3].PB(U^8);
218     // '+';
219     Pattern[4].PB(U);
220 }
```

```
221
222 int main(){
223
224 #ifndef ONLINE_JUDGE
225     freopen("/users/xiaodao/desktop/Exercise/in.txt", "r", stdin);
226     //freopen("/users/xiaodao/desktop/Exercise/out.txt", "w", stdout);
227 #endif
228
229
230    // freopen("rocketmania.in","r",stdin);
231    // freopen("rocketmania.out","w",stdout);
232
233     Init();
234
235     while (~scanf("%d", &x)){
236         OT(solve());
237         //break;
238     }
239
240 }
```

-1.6.3 World Finals – Harbin – 2009/2010 Channel

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  #define REP(i, n) for(int i=0;i<n;++i)
4  #define DWN(i, b, a) for (int i=b-1;i>=a;--i)
5  #define RST(A) memset(A, 0, sizeof(A))
6  #define FLC(A, x) memset(A, x, sizeof(A))
7  typedef long long LL;
8  const int N = 25, M = 14, _M = 3, _UM = 7;
9  char A[N][M];
10 int n, m;
11
12 int b[M+1], bb[M+1];
13 int c[M+1];
14
15 LL encode(){
16     FLC(bb, -1); int n = 1; bb[0] = 0; LL s = 0;
17     DWN(i, m+1, 0){
18         s <<= 1; s |= c[i];
19     }
20     DWN(i, m+1, 0){
21         if (!~bb[b[i]]) bb[b[i]] = n++;
22         s <<= _M; s |= bb[b[i]];
23     }
24     return s;
25 }
26 void decode(LL s){
27     REP(i, m+1){
28         b[i] = s & _UM; s >>= _M;
29     }
30     REP(i, m+1){
31         c[i] = s & 1; s >>= 1;
32     }
33 }
34
35 const int Prime = 9979, MaxSize = 1 << 18;
36 int opt[N*M+9][MaxSize], pre[N*M+9][MaxSize];
37 int i, j; LL u; int d, op;
38 struct hashTable{
39     LL state[MaxSize]; int key[MaxSize]; int sz;
40     int hd[Prime], nxt[MaxSize];
41     void clear(){
42         sz = 0;
```

```
43          FLC(hd, -1);
44      }
45      void push(){
46          int d = ::d + op; c[j] = op; LL s = encode();
47          int x = s % Prime;
48
49          //cout << " d:" << d << endl;
50
51          for (int i=hd[x];~i;i=nxt[i]) if (state[i] == s){
52              if (d > key[i]){
53                  key[i] = d;
54                  opt[::i*m+j][i] = op;
55                  pre[::i*m+j][i] = u;
56              }
57              return;
58          }
59          state[sz] = s; key[sz] = d;
60          nxt[sz] = hd[x]; hd[x] = sz;
61          opt[i*m+j][sz] = op;
62          pre[i*m+j][sz] = u;
63          ++sz;
64          assert(sz < MaxSize);
65      }
66      void roll(){
67          LL U1 = (1ll<<(_M*(m+1))) - 1;
68          LL U2 = ((1ll<<(m+1))-1) << (_M*(m+1));
69          REP(ii, sz){
70              LL s = state[ii], s1 = s & U1, s2 = s & U2;
71              s1 <<= _M; s1 &= U1;
72              s2 <<= 1; s2 &= U2;
73              state[ii] = s1 | s2;
74          }
75      }
76  } H[2]; int src, des;
77
78  void print(){
79      swap(src, des); assert(H[src].sz > 0);
80      int u = 0; for (int i=1;i<H[src].sz;++i) if (H[src].key[i] > H[src].key[u]) u = i;
81
82      int z = H[src].key[u];
83      //cout << z << endl;
84
85      DWN(i, n*m, 0){
86          int ii = i / m, jj = i % m;
87          if (opt[i][u]){
88              assert(A[ii][jj] == '.');
89              A[ii][jj] = 'C';
90          }
91          u = pre[i][u];
92      }
93      int zz = 0;
94      REP(i, n){
95          REP(j, m){
96              if (A[i][j] == 'C') ++zz;
97              putchar(A[i][j]);
98          }
99          puts("");
100     }
101     assert(z == zz);
102 }
103
104 bool c0, c1, c2;
105 void push(int _op = 0){
106     if (!_op && c0 && !c1 && c2) return; //#
107     op = _op; H[des].push();
108 }
109 int ii, lt, up; bool dn, rt; void decode(){
```

```
110     decode(H[src].state[ii]); d = H[src].key[ii]; u = ii;
111     lt = b[j], up = b[j+1];
112     c0 = j && c[j-1], c1 = c[j], c2 = c[j+1];
113 }
114 void display(){
115     decode(); REP(i, m+1) cout << b[i] << " "; cout << ": " << d << endl;
116 }
117
118
119 void solve(){
120     FLC(A, '#'); REP(i, n){
121         REP(j, m) scanf(" %c", &A[i][j]);
122     }
123
124     src = 0, des = 1; H[des].clear();
125     RST(b); RST(c); d = 0; i = j = 0; op = 0; push();
126
127     for (i=0;i<n;++i){
128         for (j=0;j<m;++j){
129             swap(src, des); H[des].clear();
130             dn = A[i+1][j] != '#'; rt = A[i][j+1] != '#';
131
132             if (A[i][j] == '#'){
133                 for(ii=0;ii<H[src].sz;++ii){
134                     decode();
135                     push();
136                 }
137                 continue;
138             }
139
140             if (i == 0 && j == 0 || i == n-1 && j == m-1){
141                 for(ii=0;ii<H[src].sz;++ii){
142                     decode();
143                     if (lt && up){
144
145                     }
146                     else if (lt || up){
147                         if (c0 && c2) continue;
148                         b[j] = b[j+1] = 0;
149                         push(1);
150                     }
151                     else{
152                         if (c0 || c1 || c2) continue;
153                         if (dn){
154                             b[j] = m; b[j+1] = 0;
155                             push(1);
156                         }
157                         if (rt){
158                             b[j] = 0; b[j+1] = m;
159                             push(1);
160                         }
161                     }
162                 }
163                 continue;
164             }
165
166             for(ii=0;ii<H[src].sz;++ii){
167                 decode();
168                 if (lt && up){
169                     if (lt == up){
170
171                     }
172                     else{
173                         //if (c1) continue; subtle ... useless ...
174                         b[j] = b[j+1] = 0; REP(i, m+1) if (b[i] == lt) b[i] = up;
175                         push(1);
176                     }
```

```
                }
                else if (lt || up){
                    if (c0 && c2) continue;
                    int t = lt | up;
                    if (dn){
                        b[j] = t; b[j+1] = 0;
                        push(1);
                    }
                    if (rt){
                        b[j] = 0; b[j+1] = t;
                        push(1);
                    }
                }
                else{
                    push();
                    if (c0 || c1 || c2) continue;
                    if (dn && rt){
                        b[j] = b[j+1] = m;
                        push(1);
                    }
                }
            }
        }

        H[des].roll();
    }

    print();
}

int main(){

#ifndef ONLINE_JUDGE
    freopen("in.txt", "r", stdin);
    //freopen("out.txt", "w", stdout);
#endif
    int cas = 0; while (~scanf("%d %d", &n, &m) && n){
        printf("Case %d:\n", ++cas);
        solve(); puts("");
    }
}
```

-1.6.4 HDU 3958. Tower Defence

```
#include <bits/stdc++.h>
using namespace std;
#define REP(i, n) for(int i=0;i<n;++i)
#define RST(A) memset(A, 0, sizeof(A))
#define FLC(A, x) memset(A, x, sizeof(A))
typedef long long LL;
const int N = 22, M = 12, _M = 3, _UM = 7;
char A[N][M];
int n, m;

int b[M+1], bb[M+1];
int c[M+1];

LL encode(){
    FLC(bb, -1); int n = 1; bb[0] = 0; LL s = 0;
    for (int i=m;i>=0;--i){
        s <<= 1; s |= c[i];
    }
    for (int i=m;i>=0;--i){
        if (!~bb[b[i]]) bb[b[i]] = n++;
        s <<= _M; s |= bb[b[i]];
```

```
22      }
23      return s;
24  }
25  void decode(LL s){
26      REP(i, m+1){
27          b[i] = s & _UM; s >>= _M;
28      }
29      REP(i, m+1){
30          c[i] = s & 1; s >>= 1;
31      }
32  }
33
34  const int Prime = 9979, MaxSize = 1 << 19;
35  int opt[N*M+9][MaxSize], pre[N*M+9][MaxSize];
36  int i, j; LL u; int d, op;
37  struct hashTable{
38      LL state[MaxSize]; int key[MaxSize]; int sz;
39      int hd[Prime], nxt[MaxSize];
40      void clear(){
41          sz = 0;
42          FLC(hd, -1);
43      }
44      void push(){
45          int d = ::d + op; c[j] = op; LL s = encode();
46          int x = s % Prime;
47
48          //cout << " d:" << d << endl;
49
50          for (int i=hd[x];~i;i=nxt[i]) if (state[i] == s){
51              if (d > key[i]){
52                  key[i] = d;
53                  opt[::i*m+j][i] = op;
54                  pre[::i*m+j][i] = u;
55              }
56              return;
57          }
58          state[sz] = s; key[sz] = d;
59          nxt[sz] = hd[x]; hd[x] = sz;
60          opt[i*m+j][sz] = op;
61          pre[i*m+j][sz] = u;
62          ++sz;
63          assert(sz < MaxSize);
64      }
65      void roll(){
66          LL U1 = (1ll<<(_M*(m+1))) - 1;
67          LL U2 = ((1ll<<(m+1))-1) << (_M*(m+1));
68          REP(ii, sz){
69              LL s = state[ii], s1 = s & U1, s2 = s & U2;
70              s1 <<= _M; s1 &= U1;
71              s2 <<= 1; s2 &= U2;
72              state[ii] = s1 | s2;
73          }
74      }
75  } H[2]; int src, des;
76
77  void print(){
78      swap(src, des); assert(H[src].sz > 0);
79      int u = 0; for (int i=1;i<H[src].sz;++i) if (H[src].key[i] > H[src].key[u]) u = i;
80
81      int z = H[src].key[u];
82      cout << z << endl;
83
84      for (int i=n*m-1;i>=0;--i){
85          int ii = i / m, jj = i % m;
86
87          //cout << "opt: " << i << " " << u << " " << pre[i][u] << " " << opt[i][u] << endl;
88
```

```
89          if (!opt[i][u]){
90              if (A[ii][jj] == '.') A[ii][jj] = 'W';
91          }
92          u = pre[i][u];
93      }
94      int zz = 2;
95      REP(i, n){
96          REP(j, m){
97              if (A[i][j] == '.') ++zz;
98              putchar(A[i][j]);
99          }
100         puts("");
101     }
102     assert(z == zz);
103 }
104
105 void push(int _op = 0){
106     op = _op; H[des].push();
107 }
108 int ii, lt, up; bool dn, rt; bool c0, c1, c2; void decode(){
109     decode(H[src].state[ii]); d = H[src].key[ii]; u = ii;
110     lt = b[j], up = b[j+1];
111     c0 = j && c[j-1], c1 = c[j], c2 = c[j+1];
112 }
113 void display(){
114     decode(); REP(i, m+1) cout << b[i] << " "; cout << ": " << d << endl;
115 }
116
117
118 void solve(){
119     scanf("%d %d", &n, &m);
120     FLC(A, 'B'); REP(i, n){
121         REP(j, m) scanf(" %c", &A[i][j]);
122     }
123
124     src = 0, des = 1; H[des].clear();
125     RST(b); RST(c); d = 0; i = j = 0; op = 0; push();
126
127     for (i=0;i<n;++i){
128         for (j=0;j<m;++j){
129             swap(src, des); H[des].clear();
130             dn = A[i+1][j] != 'B'; rt = A[i][j+1] != 'B';
131
132             if (A[i][j] == 'B'){
133                 for(ii=0;ii<H[src].sz;++ii){
134                     decode();
135                     push();
136                 }
137                 continue;
138             }
139
140             if (A[i][j] == 'S' || A[i][j] == 'T'){
141                 for(ii=0;ii<H[src].sz;++ii){
142                     decode();
143                     if (lt && up){
144
145                     }
146                     else if (lt || up){
147                         if (c0 && c2) continue;
148                         b[j] = b[j+1] = 0;
149                         push(1);
150                     }
151                     else{
152                         if (c0 || c2) continue;
153                         if (dn){
154                             b[j] = m; b[j+1] = 0;
155                             push(1);
```

```
156                         }
157                         if (rt){
158                             b[j] = 0; b[j+1] = m;
159                             push(1);
160                         }
161                     }
162                 }
163                 continue;
164             }
165 
166             for(ii=0;ii<H[src].sz;++ii){
167                 decode();
168                 if (lt && up){
169                     if (lt == up){
170 
171                     }
172                     else{
173                         //if (c1) continue; // subtle, useless...
174                         b[j] = b[j+1] = 0; REP(i, m+1) if (b[i] == lt) b[i] = up;
175                         push(1);
176                     }
177                 }
178                 else if (lt || up){
179                     if (c0 && c2) continue;
180                     int t = lt | up;
181                     if (dn){
182                         b[j] = t; b[j+1] = 0;
183                         push(1);
184                     }
185                     if (rt){
186                         b[j] = 0; b[j+1] = t;
187                         push(1);
188                     }
189                 }
190                 else{
191 
192                     push();
193 
194                     if (c0 || c2) continue;
195                     if (dn && rt){
196                         b[j] = b[j+1] = m;
197                         push(1);
198                     }
199                 }
200             }
201         }
202 
203         H[des].roll();
204     }
205 
206     print();
207 }
208 
209 int main(){
210 
211 #ifndef ONLINE_JUDGE
212     freopen("in.txt", "r", stdin);
213     //freopen("out.txt", "w", stdout);
214 #endif
215     int T; cin >> T; REP(i, T){
216         printf("Case %d: ", i+1);
217         solve(); puts("");
218     }
219 }
```