//4 4

#@##

....

####

....

2

2 1 2 4

状态压缩 ANS = 5

struct Node {

int x, y, s, t, num;/// 状态和时间 宝物数量

friend bool operator<(Node a, Node b) { /// 优先级队列 重载

return a.t>b.t;

}

};

int bao[maxn][maxn];// 记录每个宝物的编号

int vit[maxn][maxn][50];

int n, m, sx, sy, ex, ey, k, l; int ans = inf;

int dx[] = {1, -1, 0, 0}, dy[] = {0, 0, -1, 1};

char mp[maxn][maxn];

bool check(int x, int y) {

if( x >= 0 && x < n && y >= 0 && y < n && mp[x][y] != '#' ) return true;

return false;

}

void bfs() {

priority\_queue<Node> q;

/// priority\_queue<int,vector<int>,greater<int> >que3;////最小值优先 默认最大

while( !q.empty() ) q.pop();

q.push( Node{sx, sy, 0, 0, 0});

vit[sx][sy][0] = 0;

while( !q.empty() ) {

Node top = q.top();

///Node top = q.front();

q.pop();

int x = top.x, y = top.y, s = top.s, t = top.t, num = top.num;

rep(i, 0, 4) {

int nx = x + dx[i], ny = y + dy[i];

if( !check(nx, ny) ) continue;

Node next = top;

next.t = t + 1;

if( bao[nx][ny] != -1 && (next.s & (1<< bao[nx][ny]) ) == 0)

{

//cout << bao[nx][ny] << ' ' << (1<<bao[nx][ny]) << ' ' << next.s <<endl;

next.s |= (1<< bao[nx][ny]); /// 001 011

next.num++;

if( next.num == k)

{ //cout << nx << ' ' << ny << ' ' << tt << ' ';

ans = min(ans, next.t);

return;

}

}

if( vit[nx][ny][next.s] > next.t)

{

vit[nx][ny][next.s] = next.t;

q.push(Node{nx, ny, next.s, next.t, next.num});

}}}}

void init() {//初始化数组，时间为无穷大

rep(i, 0, n)rep(j, 0, m)rep(k, 0, l) vit[i][j][k] = inf;}

for(int i = 0; i < n; i++) {

scanf("%s", mp[i]);

for(int j = 0; j < m; j++) {

if( mp[i][j] == '@')

sx = i, sy = j;

}

}

int main()

{

int x, y;

while(~scanf("%d%d", &n, &m), n+m) {

mem(vit, 0);ans = inf;

int cnt = 0;

for(int i = 0; i < n; i++) {

scanf("%s", mp[i]);

for(int j = 0; j < m; j++) {

if( mp[i][j] == '@')

sx = i, sy = j;

}

}

scanf("%d", &k);

mem(bao, -1);

int i;

for(i = 0, l = 1; i < k; i++) {

scanf("%d %d", &x, &y);

bao[x-1][y-1] = i;

l = l << 1; /// 状态数 100

/////最多4个宝物，故状态的最大值为1111 l为总的状态数加一

//cout << x-1 << ' ' <<y-1 <<' ' << i << ' ' << l<<endl;

}

init();

bfs();

if( ans != inf) printf("%d\n", ans);

else printf("-1\n");

}

}

**// f.三维状压.另一种构造**

struct Node {

int x, y, t, key, snake;

};

bool vit[maxn][maxn][10][40];

int n, m, sx, sy, ex, ey;

int dx[] = {1, -1, 0, 0}, dy[] = {0, 0, -1, 1};

char mp[maxn][maxn];

bool check(int x, int y) {

if( x >= 0 && x < n && y >= 0 && y < n && mp[x][y] != '#' ) return true;

return false;

}

void bfs() {

int ans = inf;

mem(vit, 0);

queue<Node> q;

while( !q.empty() ) q.pop();

q.push( Node{sx, sy, 0, 0, 0});

while( !q.empty() ) {

Node top = q.front();

q.pop();

int x = top.x, y = top.y, t = top.t, key = top.key, snake = top.snake;

if( key == m && mp[x][y] == 'T')

{ans = min(ans, t); continue;}

if( vit[x][y][key][snake] != 0)

continue;

vit[x][y][key][snake] = 1;

rep(i, 0, 4) {

int nx = x + dx[i], ny = y + dy[i];

if( !check(nx, ny) ) continue;

Node NEXT = top;// 不能用top top四次更新 会覆盖原来的

/// if( bao[nx][ny] != -1 && (next.s & (1<< bao[nx][ny]) ) == 0)

/// next.s |= (1<< bao[nx][ny]);

/// next.num++; if( next.t == k) deal with

if( mp[nx][ny] >= 'A' && mp[nx][ny] <= 'G') ///five shakes

{

int s = mp[nx][ny] - 'A';

if( ( 1 << s ) & NEXT.snake ) ;// 如果蛇被打了

else {

//cout << NEXT.snake <<' ' << (1<<s) <<' '; 0 & 1 = 0

NEXT.snake |= ( 1<<s ); // 没被打 NEXT.snake |= ( 1<<s )

//cout << NEXT.snake << endl;

NEXT.t++;

}

}

else if( mp[nx][ny] - '0' == key + 1) /// key

NEXT.key ++;

NEXT.t++;

q.push(Node{nx, ny, NEXT.t, NEXT.key, NEXT.snake});

}

}

if( ans != inf) printf("%d\n", ans);

else printf("impossible\n");

}

int main()

{

while(~scanf("%d%d", &n, &m), n+m) {

int cnt = 0;

for(int i = 0; i < n; i++)

scanf("%s", mp[i]);

for(int i = 0; i < n; i++) {

for(int j = 0; j < n; j++) {

if(mp[i][j] == 'K') sx = i, sy = j;

if(mp[i][j] == 'S') {mp[i][j] = cnt+'A'; cnt++;}

}

}

bfs();}}

3 2

K#T

.S.

21. ANS = 8