**STATE UNIVERSITY OF BANGLADESH (SUB)**

****

**Course No: CSE-0408**

**Course Name: Artificial Intelligence Lab**

**Semester: Summer 2021**

**Submitted to:**

**Khan Md. Hasib**

**Lecturer,**

**Department of CSE, SUB**

**Submitted By:**

**Name: Sabrina Zaman Oyshe**

**ID: UG02-44-17-002**

**Batch: 44**

**Email:** sabrinaoyshe188@gmail.com

#include<bits/stdc++.h>

using namespace std;

#define D(x) cerr<<\_\_LINE\_\_<<" : "<<#x<<" -> "<<x<<endl

#define rep(i,j) for(int i = 0; i < 3; i++) for(int j = 0; j < 3; j++)

#define PII pair < int, int >

typedef vector<vector<int>> vec2D;

const int MAX = 1e5+7;

int t=1, n, m, l, k, tc;

int dx[4] = {0, 0, 1, -1};

int dy[4] = {1, -1, 0, 0};

vec2D init{

{8, 1, 2},

{3, 6, 4},

{0, 7, 5}

};

vec2D goal{

{1, 3, 2},

{8, 0, 4},

{7, 6, 5}

};

//vec2D init{

// {1, 2, 3},

// {8, 6, 0},

// {7, 5, 4}

//};

//vec2D goal{

// {1, 2, 3},

// {8, 0, 4},

// {7, 6, 5}

//};

//vec2D init{

// {1, 3, 2},

// {4, 0, 7},

// {6, 5, 8}

//};

//vec2D goal{

// {0, 2, 4},

// {1, 3, 8},

// {6, 5, 7}

//};

struct Box {

vec2D mat{ { 0,0,0 },{ 0,0,0},{ 0,0,0} };

int diff, level;

int x, y;

int lastx, lasty;

Box(vec2D a,int b = 0, int c = 0, PII p = {0,0}, PII q = {0,0}) {

rep(i,j) mat[i][j] = a[i][j];

diff = b;

level = c;

x = p.first;

y = p.second;

lastx = q.first;

lasty = q.second;

}

};

bool operator < (Box A, Box B) {

if(A.diff == B.diff) return A.level < B.level;

return A.diff < B.diff;

}

int isEqual(vec2D a, vec2D b) {

int ret(0);

rep(i,j) if (a[i][j] != b[i][j]) ret--;

return ret;

}

bool check(int i, int j) {

return i>=0 and i<3 and j>=0 and j<3;

}

void print(Box a) {

rep(i,j)

cout << a.mat[i][j] << (j == 2 ? "\n" : " ");

D(-a.diff);

D(-a.level);

cout << "(" << a.x << "," << a.y <<")\n\n";

}

void dijkstra(int x, int y) {

map < vec2D, bool > mp;

priority\_queue < Box > PQ;

int nD = isEqual(init, goal);

Box src = {init, nD, 0, {x,y}, {-1,-1}};

PQ.push(src);

int state = 0;

while(!PQ.empty()) {

state++;

Box now = PQ.top();

PQ.pop();

print(now);

if(!now.diff) {

puts("Goal state has been discovered");

cout << "level : " << -now.level << "\n";

D(state);

break;

}

if(mp[now.mat]) continue;

mp[now.mat] = true;

for(int i = 0; i < 4; i++) {

int xx = now.x + dx[i];

int yy = now.y + dy[i];

if(check(xx, yy)) {

if(now.lastx == xx and now.lasty == yy) continue;

Box temp = now;

swap(temp.mat[temp.x][temp.y], temp.mat[xx][yy]);

temp.diff = isEqual(temp.mat, goal);

temp.level = now.level - 1;

temp.x = xx;

temp.y = yy;

temp.lastx = now.x;

temp.lasty = now.y;

PQ.push(temp);

}

}

}

}

signed main() {

puts("Current State:");

rep(i,j) cout << init[i][j] << (j == 2 ? "\n" : " ");

puts("");

puts("Goal State:");

rep(i,j) cout << goal[i][j] << (j == 2 ? "\n" : " ");

puts("\n............Search Started...............\n");

rep(i,j) if(!init[i][j]) dijkstra(i,j);

return 0;

}







