Assignment-01,02: Title :Heuristic Function And Breadth-First Search(BFS)

CSE-0408 Summer 2021

Name:Abu Noman Farhad
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
email address:ahamedfarhad1@gmail.com

***Abstract*—A heuristic function h(n) , takes a node n and returns a non-negative real number that is an estimate of the cost of the least-cost path from node n to a goal node. The function h(n) is an admissible heuristic if h(n) is always less than or equal to the actual cost of a lowest-cost path from node n to a goal.**

***Index Terms*—About Heuristic Function ,the 8-puzzle problem in Python.**

I. Introduction

Definition:The heuristic function is a way to inform the search about the direction to a goal. It provides an informed way to guess which neighbor of a node will lead to a goal. There is nothing magical about a heuristic function. It must use only information that can be readily obtained about a node. Objective of heuristics function:The heuristic function is a way to inform the search about the direction to a goal. It provides an informed way to guess which neighbor of a node will lead to a goal. There is nothing magical about a heuristic function. It must use only information that can be readily obtained about a node

II. LITERATURE REVIEW

The study of heuristics in human decision-making was developed in the 1970s and the 1980s by the psychologists Amos Tversky and Daniel Kahneman although the concept had been originally introduced by the Nobel laureate Herbert A. Simon, whose original, primary object of research was problem solving that i showed that..

III. PROPOSED METHODOLOGY

Heuristics are methods for solving problems in a quick way that delivers a result that is sufficient enough to be useful given time constraints. Investors and financial professionals use a heuristic approach to speed up analysis and investment decisions. Current state [[8,1,2],[3,6,4],[0,7,5]] and the goal state is [[1,2,3],[8,0,4],[7,6,5

IV. RULES OF SOLVING PROBLEM

It is represented by h(n), and it calculates the cost of an optimal path between the pair of states. The value of the heuristic function is always positive. Admissibility of the heuristic function is given as: h(n) ¡= h*(n) 1.UP

2.Down

3.Right

4.Left

V. Code-1

```
include¡bits/stdc++.h¿
using namespace std;
define D(x) cerr¡¡_LINE_<<":"<<x<<"->"<<x<<endl
define rep(i,j) for(inti = 0; i¡ 3; i++) for(int j = 0; j ¡ 3; j++)
define PII pair ¡int, int¿ typedef vector¡vector¡int¿¿vec2D;
constint MAX = 1e5+7;
int t=1, n, m, l, k, tc;
intdx[4] = 0, 0, 1, -1;
intdy[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;
intlastx, 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;
intisEqual(vec2D a, vec2D b)  intret(0);
rep(i,j) if (a[i][j] != b[i][j]) ret–; return ret;
```

```
bool check(inti, 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 ? "" : " ");
D(-a.diff);
D(-a.level);
cout¡¡ "(" ¡¡a.x¡¡ "," ¡¡a.y¡¡")";
void dijkstra(int x, int y)  map ¡ vec2D, bool¿mp;
priority_queue < Box > PQ;
intnD = 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¡¡ "";
D(state);
break;
if(mp[now.mat]) continue;
mp[now.mat] = true;
for(inti = 0; i¡ 4; i++)  int xx = now.x + dx[i];
intyy = 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 ? "" : " ");
puts("");
puts("Goal State:");
rep(i,j) cout¡¡ goal[i][j] ¡¡ (j == 2 ? "" : " ");
puts(".....Search Started.....");
rep(i,j) if(!init[i][j]) dijkstra(i,j);
return 0;
```

VI. Output

Fig. 1. Example of a Output figure

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989. York: Academic, 1963, pp. 271–350.

[4] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

***Abstract*—This paper introduced for Breadth-First Search(BFS) problem solved using C++ language.**

***Index Terms*—About Heuristic Function ,the 8-puzzle problem in Python.**

Index Terms—Here I mostly used in My report C++ language and Code-block code editor.

VII. Introduction

Breadth-first search starts at a given vertex s, which is at level 0. In the first stage, we visit all the vertices that are at the distance of one edge away. When we visit there, we paint as "visited," the vertices adjacent to the start vertex s - these vertices are placed into level 1. In the second stage, we visit all the new vertices we can reach at the distance of two edges away from the source vertex s. These new vertices, which are adjacent to level 1 vertices and not previously assigned to a level, are placed into level 2, and so on. The BFS traversal terminates when every vertex has been visited.

VIII. Literature Review

BFS and its application in finding connected components of graphs were invented in 1945 by Konrad Zuse, in his (rejected) Ph.D. thesis on the Plankalk ul programming language, but this was not published until 1972. It was reinvented in 1959 by Edward F. Moore, who used it to find the shortest path out of a maze,[5][6] and later developed by C. Y. Lee into a wire routing algorithm (published 1961). In 2012 Farhad S. et. al. [4] proposed new resolution for solving N-queens by using combination of DFS (Depth First Search) and BFS (Breadth First Search) techniques

IX. Proposed Methodology

Here i Discuss BFS Algorithm:

1. for each u in V s
2. do color[u] ←WHITE
3. d[u] ←infinity
4. [u] ←NIL
5. color[s] ←GRAY
6. d[s] ←0
7. [s] ←NIL
8. Q ←
9. ENQUEUE(Q, s)
10 while Q is non-empty
11. do u ←DEQUEUE(Q)
12. for each v adjacent to u
13. do if color[v] ←WHITE
14. then color[v] ←GRAY
15. d[v] ←d[u] + 1
16. [v] ←u
17. ENQUEUE(Q, v)
18. DEQUEUE(Q)
19. color[u] ←BLACK

Fig. 2. Example of a Output figure

X. SOLVING METHOD

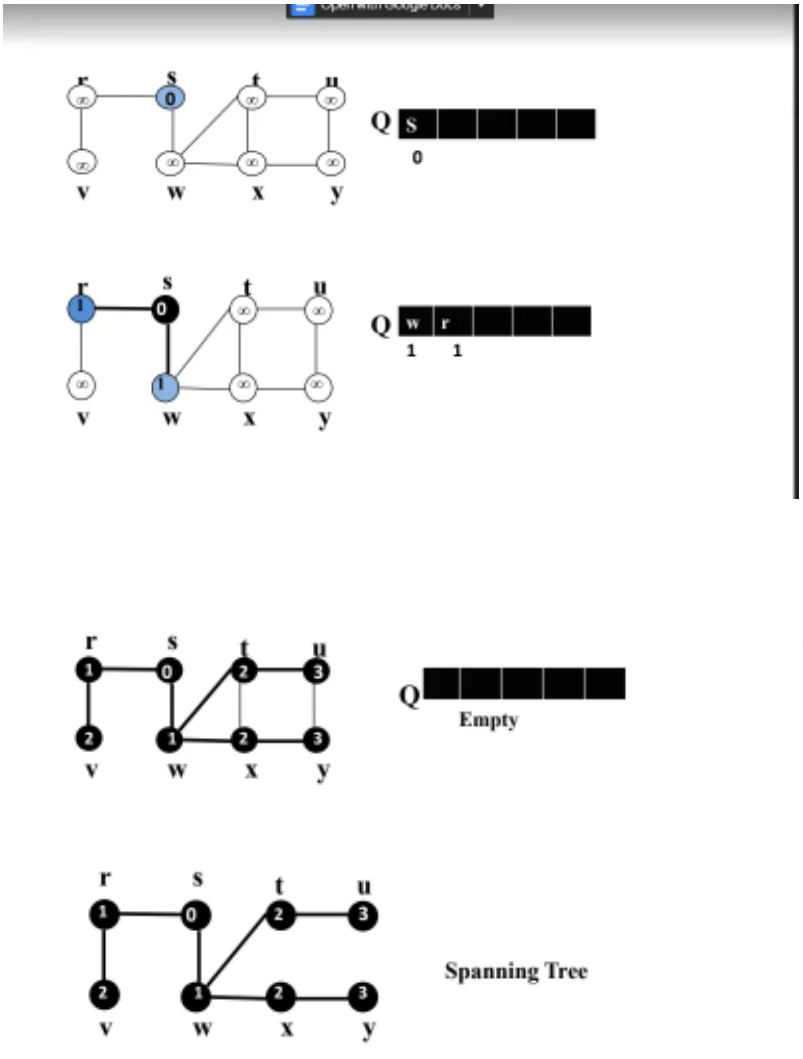
XI. CODE-2

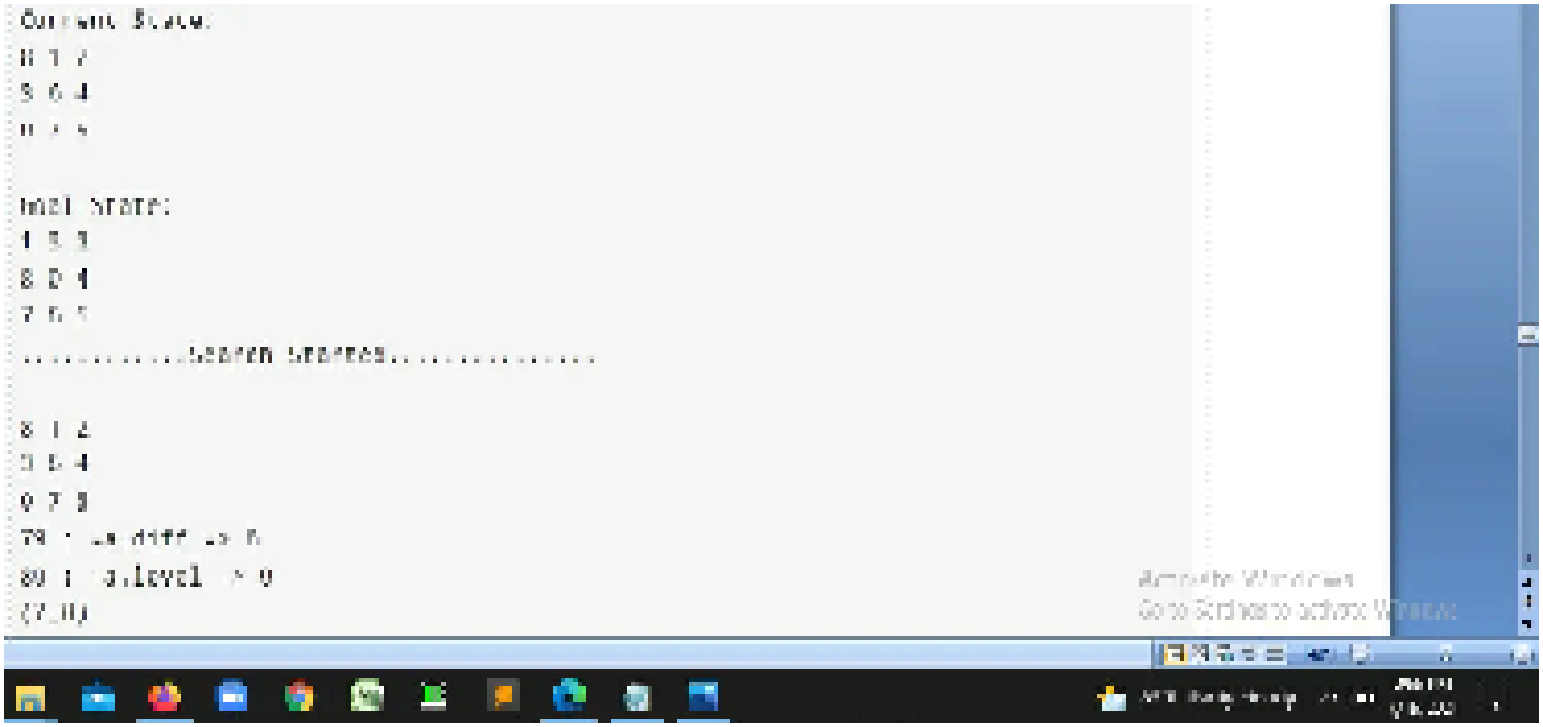
```
include ¡bits/stdc++.h¿
using namespace std;
int V,E;
int parent[100];
int cost [1000][10000];
int find(int i)
while (parent[i] != i)
i = parent[i];
return i;
void union1(int i, int j)
int a = find(i);
int b = find(j);
parent[a] = b;
void BFsMST()
int mincost = 0;
int edge_count = 0;
while (edge_count < V − 1)
int min = INT_MAX, a = −1, b = −1;
for (int i = 0; i ¡ V; i++)
for (int j = 0; j ¡ V; j++)
if (find(i) != find(j)  cost[i][j] ¡ min)
min = cost[i][j];
a = i;
b = j;
union1(a, b);
cout¡¡"Edge "¡¡edge_count++ << " : (" << a << "" <<
b << ")
cost:"¡¡min¡¡endl;
mincost += min;
cout¡¡endl¡¡"Minimum cost= "¡¡mincost;
int main()
//freopen("input.txt","r",stdin);
//cin¿¿V¿¿E;
V=8;
E=9;
for(int i=0;i¡V;i++)
for(int j=0; j¡V; j++)
cost[i][j]= INT_MAX;
/*for(int j = 0;j¡E;j++)
int m,n;
cin¿¿m¿¿n;
cin¿¿cost[m][n];
//cost[n][m];
*/
cost[0][1]=1;
cost[0][2]=2;
cost[2][3]=3;
cost[3][5]=5;
cost[3][4]=4;
cost[5][4]=100;
cost[5][6]=7;
cost[6][7]=101;
cost[4][7]=8;
for (int i = 0; i ¡ V; i++)
parent[i] = i;
/// Print the solution
BFsMST();
return 0;
```

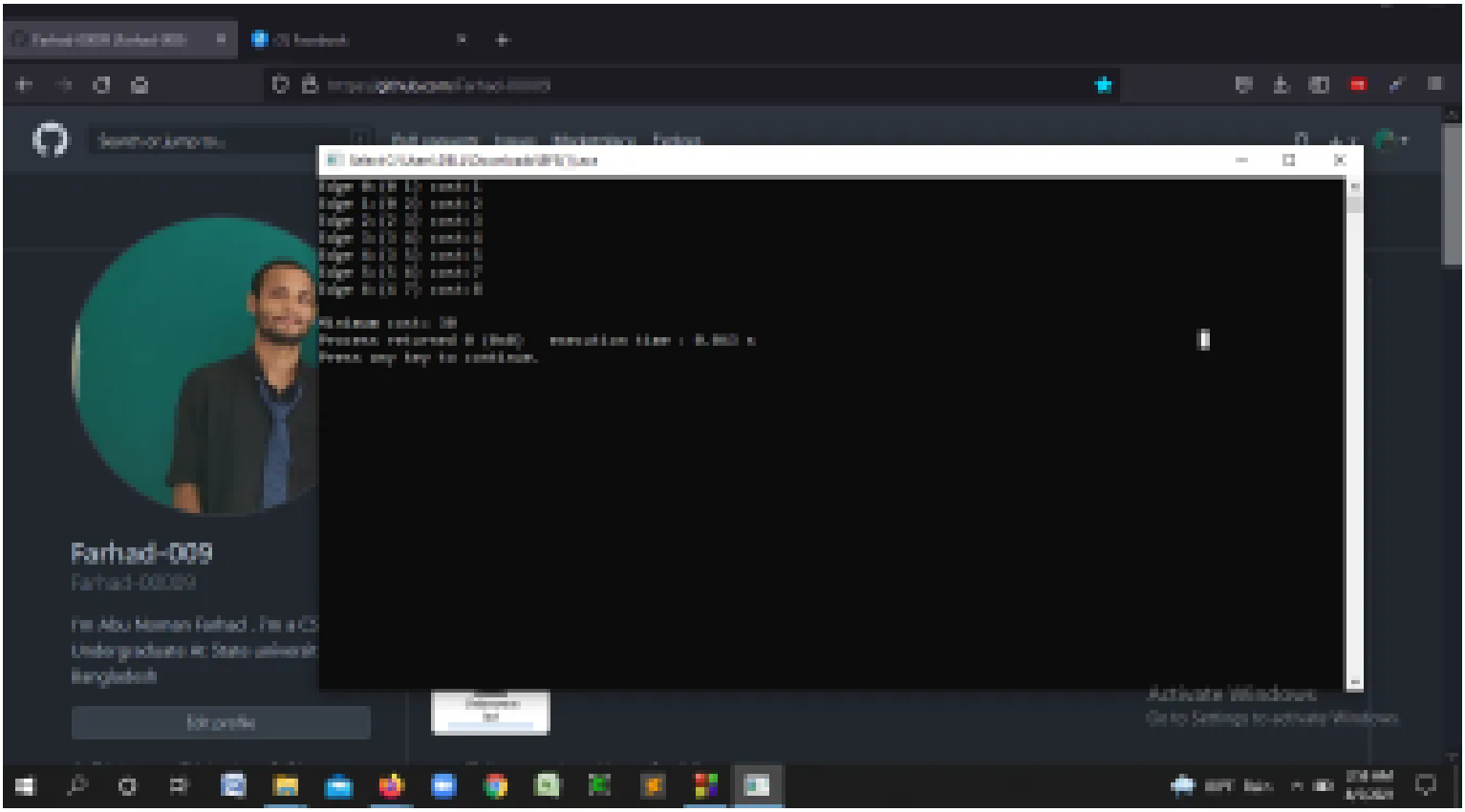
XII. OUTPUT- 1x2

ACKNOWLEDGMENT

I Would Like To Like Thanks To My Honorable Khan Md.Hasib Sir For This Time.




Current State:
Goal state:
Activate Windows
Go to Settings to activate Windows.


Farhad-009
Activate Windows
Go to Settings to activate Windows.