Ibrahim Alizada - AI Project 1

Algorithm choice : I used Dijkstra algorithm (UCS) for uninformed search and A* for informed one.

How it applies to the problem :

For uninformed search, it simply applies Dijkstra logic. For each "source" node case, algorithm searches the adjacent nodes and find the shortest distance to them, and then search next "source" node with the lowest distance.

For informed search, I just added heuristic function using Euclidean distance formula to Dijkstra (distances between the corresponding squares * 500).

Note : In normal, we had to multiply heuristic values by 100 to normalize edge distances. But, I multiplied by 500 (heuristic function * 500) to get better time and space complexity and admissible results.

How to run my script : I implemented the task in Python using Jupyter Notebook. It reads e.txt and v.txt files and accepts 1 input : Source and Destination nodes -> Example : 0 99

It gives back the results for informed and uninformed search separately and compare their performance measures in the end.

```
Enter source and destination vertex : 0 99


The results using  uninformed search (UCS) :

Path : 99 <- 0

Fringes count:  57
Fringes: 0,9,73,59,2,24,40,31,23,38,36,61,13,29,76,47,32,57,8,77,50,30,67,19,34,72,71,22,88,64,82,91,6,68,90,28,63,5
2,53,18,1,41,12,98,49,39,62,69,87,55,21,3,35,92,93,15,99

The cost from  Source Node ( 0 )  to ->  Destination Node( 99 )  is :  6157
Time required to implement :  0.009000778198242188




The results using   informed search (A*) :

Path : 99 <- 0

Fringes count:  24
Fringes: 0,9,24,73,2,40,76,23,59,31,36,82,57,61,8,52,38,64,18,6,1,47,28,99

The cost from  Source Node ( 0 )  to ->  Destination Node( 99 )  is :  6157
Time required to implement :  0.005983591079711914



Performance comparison :

A* searches 33 fringes less : 2.4  times  less

A* searches 0.00302 seconds less : 1.50  times  faster
```