



Faculty of Engineering and Technology
Department of Electrical and Computer Engineering
Artificial Intelligence ENCS 3340
Second Semester, 2021/2022

PROJECT#1 Search Algorithms for Route Navigation

Prepared by: Osama Qutait

Student's No.: 1191072

Partner's name: Mahmoud Samara

Partner's No.: 1191602

Instructor: Dr. Adnan Yahya

Date: 17-5-2022

.....

Abstract

First of all, this will be an implementation of search algorithms for a specific goal of finding an optimal path between cities in Palestine. So we need at first choose 20 cities (Historical Palestine). The cities me and my partner chose are: (**Aka, Bethlehem, Dura, Haifa, Halhoul, Hebron, Jenin, Jericho, Jerusalem, Nablus, Nazareth, Qalqilya, Ramallah, Ramleh, Sabastia, Safad, Salfit, Tubas, Tulkarm, Yafa**). After that, we will make an excel file and we have to enter for each pair cities 3 values and the values are: (Car distance (km), Aerial distance (km), Walking Distance (KM)). But note that the car distance can't be for all pairs because we can arrive to a city from other cities, and the same thing for the walk distance in some algorithms we will design the graph and it. Now we are ready to design our graph that will be our reference for the algorithms the directed edge between nodes($A \rightarrow B$) represents the existence of a path between A and B and the link is labelled by the cost of moving from A to B by car (CarDist(A,B). Also we made our graph undirected so we can go from A to B and from B to A.

Since the team numbers are 1191072, 1191602, So $2 \text{ mode } 4 = 2$ then we will take group number 2. We will make the following algorithms:

1. Breadth First Search (BFS)
2. Uniform Cost Search (UCS)
3. A* search (Aerial (heuristic) and Walk (real cost))
4. A* search (Walk (heuristic) and Car (real cost))
5. For the bonus part we will make extra algorithm that is Greedy

We will use python language for our project and from excel file that will be attached with the project folder, we will make from it 3 input files since we have cases and different inputs depend on the algorithm type.

Table of Contents

1. Program Implementation	4
2. Running Example	5
3. Bonus Elements	9

List of Figures

Figure 1-1 Project content.....	4
Figure 2-1 Uniform cost example 1	5
Figure 2-2 Uniform cost example 2	5
Figure 2-3 Uniform cost example UI.....	6
Figure 2-4 A* example 1.....	7
Figure 2-5 A* example 1-1	7

1. Program Implementation

We used python to implement our program using pycharm 2021, first of all we created 3 input files and we took the numbers from the excel file. The three input files are:

1. BFS.txt (input file): this file will be used for BFS algorithm and UCS algorithm it contains only the cities that are directly connected and their cost (car cost), also we will use it to build our graph since it depend on the car cost. In this file we put city, neighbor city, real cost between them and we split depend on ## like (Nazareth##Safad##85##Qalqilya##Nablus##54##)
2. A.txt (input file): this file will be used for A* case 1 when the walk is the real cost, so any city no direct connect between them by car cost no direct connect by walk cost. In this file we put city, neighbor city, walk cost as real cost between them and we split depend on ## like (Ramleh##Yafa##19## Sabastia##Jenin##27##)
3. Data.txt (input file): this file will be used for A* case 2 when the car is the real cost, so we will make the walk as heuristic so it has 400 value and car cost just for cities connected together directly. In this file we put city, neighbor city, aerial cost, walk cost , car cost cost between each two cities and we split depend on ## like (Sabastia##Tulkarm##17##19##36 Safad##Aka##40##50##54

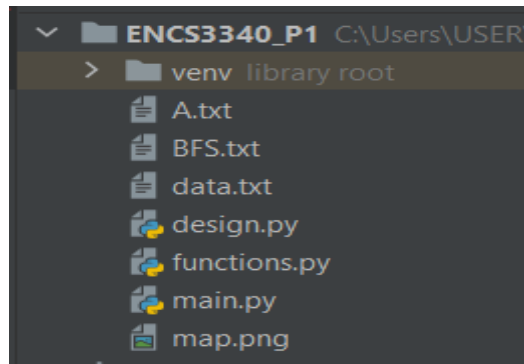


Figure 1-1 Project content

We can see that we have the 3 input files in pycharm and the main python class that contain all algorithms we did and how we read our input files.

How to run the code:

1. The user must choose number from the menu that will appear for him\her.
2. After choose the algorithm he\she must choose the start city (source).
3. Now the user must choose the number of goals he want as we made it multiple goals and the start city is the same for all goals
4. Now everything is ready and the program will print the information of every two cities for example: In BFS it will print (visited cities and best path but there is no cost since it is uninformed search algorithm), In UCS it will print (visited cities and best path and real cost (car cost)), In A* first case it will print (visited cities and best path and real cost (walk cost)) , In A* second case it will print (visited cities and best path and real cost (car cost)) , for the extra algorithm we made Greedy it will print (visited cities and best path but there is no cost since we are deal with heuristic values).

2. Running Example

For example In UCS algorithm we can see in the following figure we can see that we choose the start city that is (Nablus) and we choose two goal cities that are (Haifa and bethlehem) so we got the following information:

```
PLZ enter number of algorithm from the previous MENU
3
WELCOME to Uniform Cost Search Algorithm(UCS)

The 20 cities and how they must be wrote:
['Aka', 'Bethlehem', 'Dura', 'Haifa', 'Halhoul', 'Hebron', 'Jenin', 'Jericho', 'Jerusalem', 'Nablus', 'Nazareth', 'Qalqilya', 'Ramallah', 'Ramleh', 'Sabastia', 'Safad', 'Salfit']

PLZ enter the start city
Nablus
PLZ enter number of goal cities
2
the number of goals is: 2

PLZ enter the goal city number 1
Haifa
The visited cities from Nablus to the goal Haifa are:
Visit Nablus=> Visit Sabastia=> Visit Tubas=> Visit Salfit=> Visit Jenin=> Visit Qalqilya=> Visit Tulkarm=> Visit Ramallah=> Visit Jerusalem=> Visit Bethlehem=> Visit
=====
The real path from Nablus to the goal Haifa are: ['Nablus', 'Sabastia', 'Jenin', 'Haifa']
=====

Shortest distance between Nablus to the goal Haifa is 128
```

Figure 2-1 Uniform cost example 1

```
PLZ enter the goal city number 2
Bethlehem
The visited cities from Nablus to the goal Bethlehem are:
Visit Nablus=> Visit Sabastia=> Visit Tubas=> Visit Salfit=> Visit Jenin=> Visit Qalqilya=> Visit Tulkarm=> Visit Ramallah=> Visit Jerusalem=> Visit Bethlehem=>
=====
The real path from Nablus to the goal Bethlehem are: ['Nablus', 'Ramallah', 'Jerusalem', 'Bethlehem']
=====

Shortest distance between Nablus to the goal Bethlehem is 87
```

Figure 2-2 Uniform cost example 2

Also our code we made for it User Interface as we can see:



Figure 2-3 Uniform cost example UI

Also for A* we can see that we can choose more than one goal and it works fine:

```
PLZ enter number of algorithm from the previous MENU
6
WELCOME to A* Search Algorithm( Walk (heuristic) and Car (real cost)

The 20 cities and how they must be wrote:
['Aka', 'Bethlehem', 'Dura', 'Haifa', 'Halhoul', 'Hebron', 'Jenin', 'Jericho', 'Jerusalem', 'Nablus', 'Nazareth', 'Qalqilya', 'Ramallah', 'Ramleh', 'Sabastia', 'Safad', 'Salfit']

PLZ enter the start city
Qalqilya
PLZ enter number of goal cities
2

the number of goals is: 2

PLZ enter the goal city number 1

Safad
The visited cities from Qalqilya to the goal Safad are:
Visit Qalqilya=> Visit Sabastia=> Visit Jenin=> Visit Nazareth=> Visit Nablus=> Visit Tubas=> Visit Tubas=> Visit Tulkarm=> Visit Yafa=> Visit Aka=> Visit Safad=>
=====
The real path from Qalqilya to the goal Safad are: ['Qalqilya', 'Sabastia', 'Jenin', 'Nazareth', 'Safad']

=====

Shortest distance between Qalqilya to the goal Safad is 198
```

Figure 2-4 A* example 1

```
PLZ enter the goal city number 2

Dura
The visited cities from Qalqilya to the goal Dura are:
Visit Qalqilya=> Visit Yafa=> Visit Ramleh=> Visit Jerusalem=> Visit Sabastia=> Visit Bethlehem=> Visit Hebron=> Visit Dura=>
=====
The real path from Qalqilya to the goal Dura are: ['Qalqilya', 'Yafa', 'Ramleh', 'Jerusalem', 'Bethlehem', 'Hebron', 'Dura']

=====

Shortest distance between Qalqilya to the goal Dura is 178
```

Figure 2-5 A* example 1-1



3. Bonus Elements

For the Bonus elements:

1. We made an extra algorithm we feel of interest that is Greedy search and it is working successfully and we assumed that the heuristic values are walk values since we asked Dr. Adnan and he said you can use either aerial or walk as heuristic.
2. We made more factors for optimization like we made our graph undirected graph so you can go from A to B and from B to A this will help us to find the shortest path successfully that what we saw in UCS and A* algorithms when we ran the program. Moreover, as we can see in the excel file always the heuristic value is less than or equal the real cost this help us to make the A* admissible so it will lead to make it optimal. Also, we made error messages to avoid the stop of the running like if user enter negative value or if the start city or goal city entered in wrong way
3. Our program can let the user enter the start and the goal so they are not constant. In addition, the user can choose the number of goals he wants so our program is dealing with multiple goals and also it is multiple A* as it is bonus thing.
4. We made user interface as shown in the previous figures