جامعة بيرزيت

Department of Electrical and Computer Engineering

ENCS3340 Artificial Intelligence

Programming Project 1

Student Names: Diaa Maali & Amal Ziad

Student ID’s : 1180933 & 1192141

Instructors: Dr.Adnan Yahya &Yazan Abu Farha

Section : 3&2

Date:17-5-2022

Introduction

We have a map consist of 20 Palestinian cities, and the aim of our project is to reach goal nodes in shortest path with least cost by using the implementation of the four search algorithms: A*, Breadth and Greedy using eclipse (java language with scene builder).

In computer science, a search algorithm is any algorithm which solves the search problem, namely, to retrieve information stored within some data structure, or calculated in the search space of a problem domain, either with discrete or continuous values.

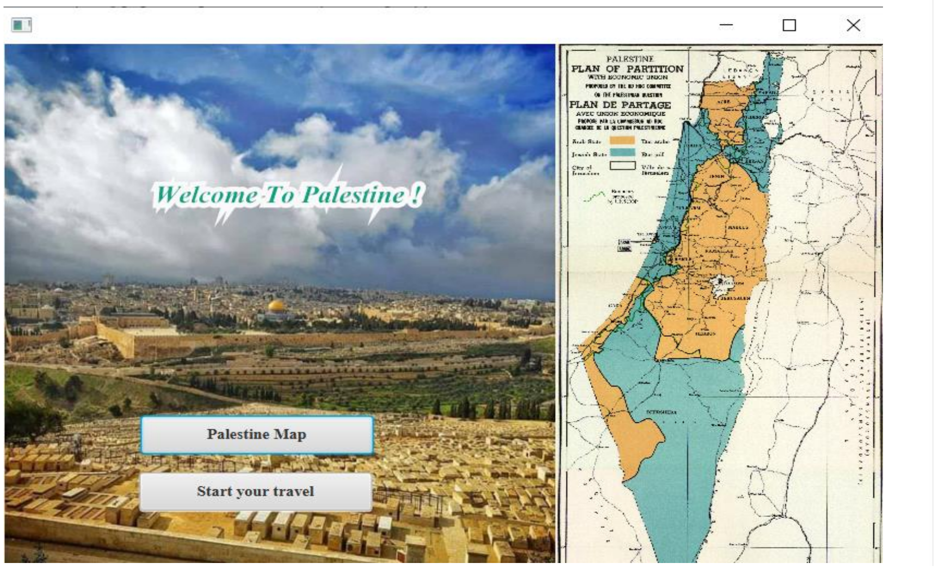
In our project we dealt with two types of search algorithm, the first one is Breadth first which is belong to the uniformed search. And the second is A* which it belong to Informed search , and the Greedy belong to Informed search.

Uninformed search is a group of wide range usage algorithms of the era. These algorithms are brute force operations, and they don’t have extra information about the search space; the only information they have is on how to traverse or visit the nodes in the tree. Thus uninformed search algorithms are also called blind search algorithms. The search algorithm produces the search tree without using any domain knowledge, which is the brute force in nature. They are different from informed search algorithms in a way that you check for a goal when a node is generated or expanded, and they don’t have any background information on how to approach the goal. The informed search algorithm is more useful for large search space.

Informed search algorithm uses the idea of heuristic to rank nodes and select the most promising one for expansion, so it is also called Heuristic search.

Procedure:

Once we run our program, the user interface appeared which shows a 2 button, as you see in the figure1 below:-



The first button for the Palestinian map, if we press on it, fig2 will appear which contains the cities and the real distances between them, and another for button that move the user to a new page for making the user define his details information for his journey by asking him to enter the start city, the goal city that he want to reach, the time to travel, and then choose one of the three algorithms to approach: (1.A* 2. Breadth First 3. Greedy) and finally the Alert message appeared which contains the shortest path of the journey that user defined with useable algorithm

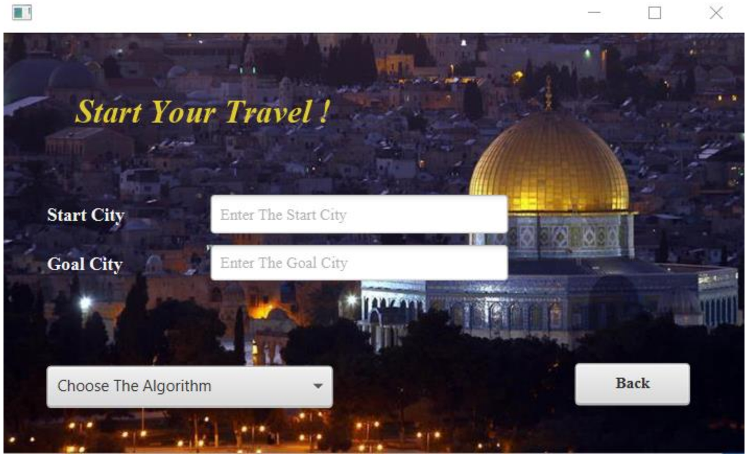


Fig3

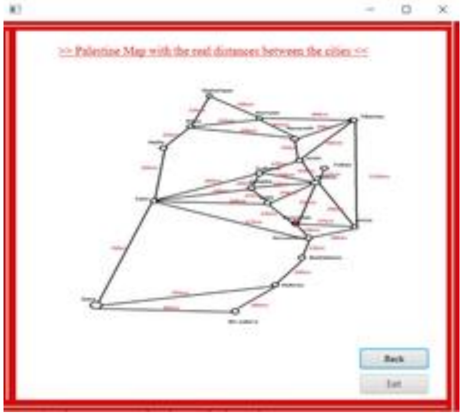

>> Palestine Map with the real distances between the cities <<
Back
Exit

Fig2

Note:If the city name is entered incorrectly, it will give a message indicating that

1. Breadth First Search :

Breadth first search is a graph traversal algorithm that starts traversing the graph from root node and explores all the neighboring nodes. Then, it selects the nearest node and explore all the unexplored nodes. The algorithm follows the same process for each of the nearest node until it finds the goal, and the fringe is a FIFO queue.

How we implement Breadth first:

The implementation uses a Queue data structure, to maintain the sequence of visiting and exploring the nodes it utilizes, because Queue depends on First in First Out.

***.

2. A* algorithm

A* search is the most commonly known form of best-first search. It uses heuristic function h(n), and cost to reach the node n from the start state g(n). A* search algorithm finds the shortest path through the search space using the heuristic function. The evaluation function f(n) is the estimated total cost of the path through node *n* to the goal: $f(n) = g(n) + h(n)$

g(n): cost so far to reach n (path cost)

h(n): estimated cost from n to goal (heuristic)

How we implement A* algorithm:

We implemented a function for calculating the evaluation function $f(n) = g(n)+h(n)$ for each accessory of the nodes, then choose the least value of f(n) (the shortest path) .

***.

3. Greedy best-first:

search with a heuristic that attempts to predict how close the end of a path is to a solution (or, goal), so that paths which are judged to be closer to a solution (or, goal) are extended first.

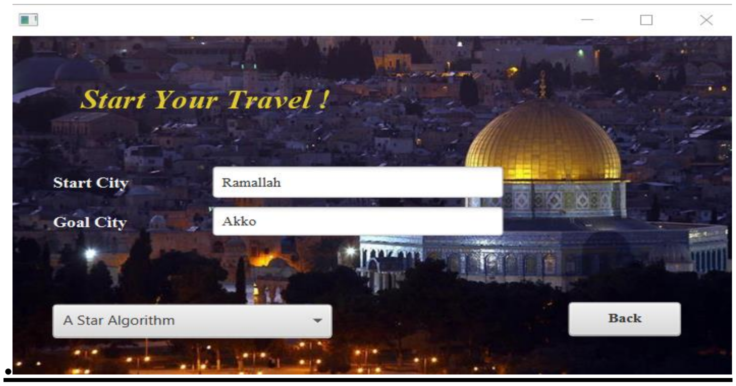
How we implement Greedy BFS algorithm:

We implemented a function for calculating the evaluation function $f(n) = h(n)$.
for each accessory of the nodes , then choose the least value of f(n) (the shortest path) .

***.

Project Ruining:

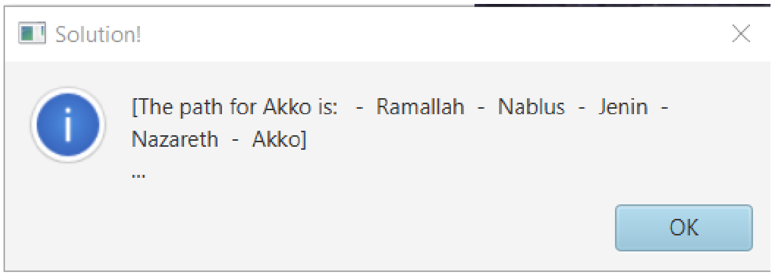
1) A* algorithm:



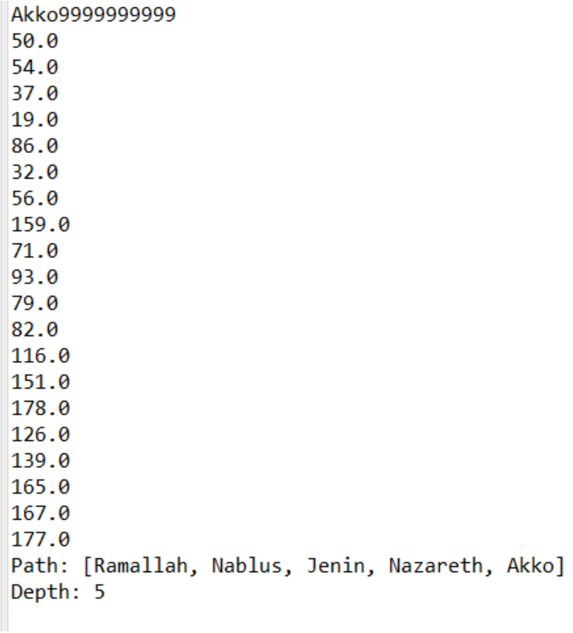
Start City Ramallah

Goal City Akko

A Star Algorithm

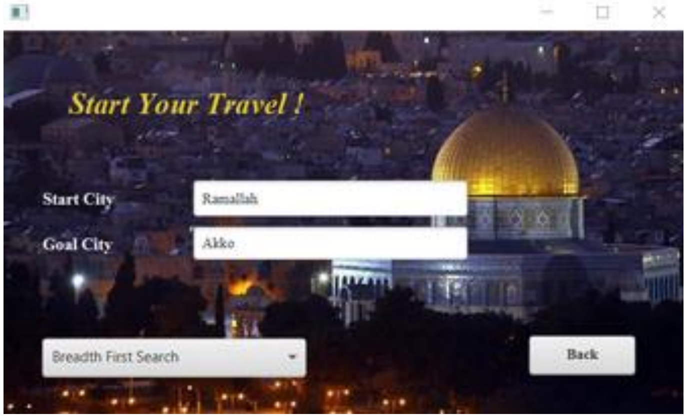


[The path for Akko is: - Ramallah - Nablus - Jenin - Nazareth - Akko]
...



```
Akko9999999999
50.0
54.0
37.0
19.0
86.0
32.0
56.0
159.0
71.0
93.0
79.0
82.0
116.0
151.0
178.0
126.0
139.0
165.0
167.0
177.0
Path: [Ramallah, Nablus, Jenin, Nazareth, Akko]
Depth: 5
```

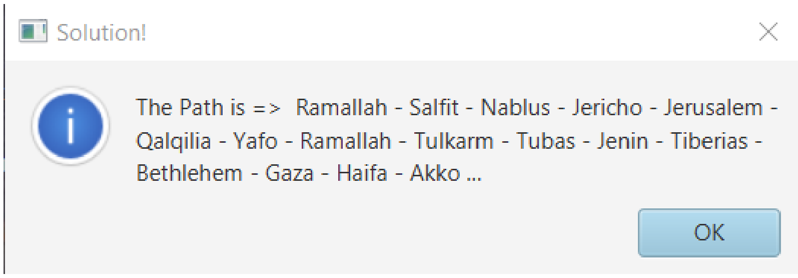
2) Breadth First Search :



Start City: Ramallah

Goal City: Akko

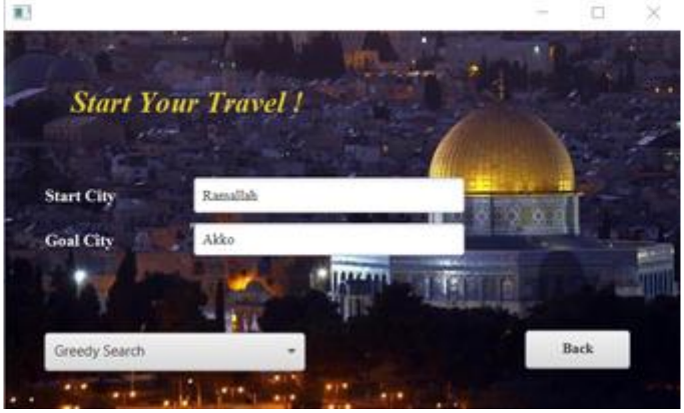
Breadth First Search



Solution!

The Path is => Ramallah - Salfit - Nablus - Jericho - Jerusalem - Qalqilia - Yafo - Ramallah - Tulkarm - Tubas - Jenin - Tiberias - Bethlehem - Gaza - Haifa - Akko ...

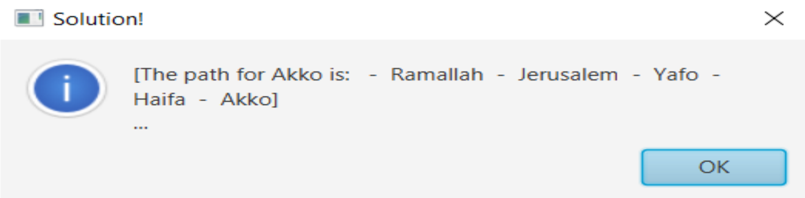
3) Greedy best-first:



Start City: Ramallah

Goal City: Akko

Greedy Search



Solution!

[The path for Akko is: - Ramallah - Jerusalem - Yafo - Haifa - Akko]
...