Project 5: A* Algorithm

CMPE 250, Data Structures and Algorithms, Fall 2016

Instructor: H. L. Akın, A. T. Cemgil
TA: Çağatay Yıldız
SA: Rıza Özçelik

Due: 18.12.2016, 23.55

1 Description

Well... We know that you are already tired of reading descriptions, we are tired of writing as well, so this will be a one sentenced description!

Implement the A* algorithm described in class. The algorithm will find the shortest path from a node to another. In contrast to Dijkstra's algorithm, **A*** considers the distance between cities as well as the heuristic values. Below are very useful links that would help you a lot in the project.

Link to the slide
Link to the video series

2 Input & Output

Input and output files will have the following format and their name will be specified via terminal as usual. For input:

- First line will contain V and E Number of cities and roads respectively.
- Next V line will contain straight line distances to the destination (i.e. Value of the heuristic functions)
- Next E line will contain three integers that will represent the bidirectional roads. The first two integers will define the vertices and the last one will define the length of the road.
- Last line will have two integers that are the ids of the source and the destination.

For output just print one integer that is the shortest distance between two cities. (The table 1 is on the last page)

Note that city enumeration is done as follows for the Romania map in videos and slides:

0. Arad
1. Zerind
2. Oradea
3. Sibiu
4. Timisoara
5. Lugoj
6. Mehadia
7. Dobreta
8. Craiova
9. Rimnicu Vicea
10. Pilesti
11. Fagaras
12. Neamt
13. Iasi
14. Vaslui
15. Urziceni
16. Bucharest
17. Giurgiu
18. Hirsova
19. Eforie

Submission Details

You are supposed to use the Git system provided to you for all projects. No other type of submission will be accepted. Also pay attention to the following points:

- Make sure that an executable is created after executing `cmake CMakeLists.txt` and `make` commands. Otherwise, you will get no credit. Your code will be tested with the command:

  ```
  ./project5 inputFile outputFile
  ```

- All source codes are checked automatically for similarity with other submissions and exercises from previous years. Make sure you write and submit your own code.

- In our case, you are expected to use C++ as powerful, steady and flexible as possible. Use mechanisms that affects these issues positively.

- Make sure you document your code with necessary inline comments, and use meaningful variable names. Do not over-comment, or make your variable names unnecessarily long.

Sample Input File	Sample Output File
20 23	418
366	
374	
380	
253	
329	
244	
241	
242	
160	
193	
10	
176	
234	
226	
199	
80	
0	
77	
151	
161	
0 1 75	
0 3 140	
0 4 118	
1 2 71	
2 3 151	
4 5 111	
5 6 70	
6 7 75	
7 8 120	
8 9 146	
3 9 80	
3 11 99	
9 10 97	
8 10 138	
10 16 101	
11 16 211	
16 17 90	
12 13 87	
13 14 92	
14 15 142	
15 16 85	
15 18 98	
18 19 86	
0 16	

Table 1: Representation of the graph in the slides.