CIS 4930 INTRODUCTION TO AI

A* Search Algorithm
DUE: 9/24/2018, 11:59 PM

DESCRIPTION

Write a program which implements the A* algorithm on an arbitrary map described through "connections" file and "locations" file. The "locations" file includes the names of the locations and their x and y positions. The "connections" file includes how the locations are connected. The algorithm should allow the user to exclude 1 or more cities from the solution path.

Two A* algorithm heuristics, "Straight Line Distance" and "Fewest Cities" should be implemented and the user should select which one should be used for a given run. Note: The "Fewest Cities" is the path in which the fewest number of cities are included in the solution path.

The program should also allow the user to select whether to show just the optimal path found or a "step-by-step" path to the solution ending with the optimal path. This "step-by-step" option should show how the A* algorithm is progressing through the search. At every step it should show:

1. **The name of the city** selected from which to search next.
2. **From the city selected**, name all the possible cities to where to travel next.
3. **After all possible paths are evaluated**, list all the cities at the end of paths that are still being considered in the search and the estimated total path distance.

The user input at runtime consists of the following:

- The start city
- The end city
- The city name(s) that should not be included in the solution path; let the user select 1 or more cities that are to be excluded.
- "step-by-step" option or not.
- The heuristic to use
    - Straight line distance, or
    - Fewest Cities

OUTPUT

If the "step-by-step" option is NOT selected, the program output should be the solution path from start city to end city as a series of segments. Each segment should include its length (in distance or number of cities depending on the heuristic selected). The total length of the path should also be included at the end. – for example if the start city is B2 and the end city is E4 the output might be:

B2 to C2 length 3.4, or B2 to C2 length 2

```
C2 to C3 length 4.8,   or C2 to C3 length 1
C3 to C4 length 1.5,   or C3 to C4 length 1
C4 to D4 length 7.1,   or C4 to D4 length 1
D4 to E4 length 3.7    or D4 to E4 length 1

Total path length: 20.5  or Total path length: 6
```

<u>If the "step-by-step" option IS selected</u>, the program output, prior to showing the solution path as above, should provide the following information at every step. For example:

```
City selected: B2

Possible cities to where to travel: C2 C3 A2 B1

Cities at the end of possible paths: C2(3.4)  C3(3.1) A2(4.7)  B1(7.3)

**************************************************************

City selected: C3

Possible cities to where to travel:  C2 B2 C4

Cities at the end of possible paths: C2(3.4)  C2(6.4)  A2(4.7)  B1(7.3)
```

DO NOT INCLUDE

ETC

```
*****************************************************************
The final solution path is:

B2 to C2 length 3.4,
C2 to C3 length 4.8,
C3 to C4 length 1.5,
C4 to D4 length 7.1,
D4 to E4 length 3.7

Total path length:20.5
```

SUBMISSION

Through a link in Canvas. For those working in pairs, perform a single submission.

DELIVERABLES

Assignment should be submitted as a single ZIP file which include source files, and a readme text file with your and your partner's name and instructions about how to compile and execute the program from a command line.

PROGRAMMING GUIDELINES

You are allowed to use the programming language of your preference. For grading purposes, the use of the following languages is recommended: Python, Java, C (gcc), C++ (g++), Javascript. For other languages, you will need to schedule an appointment with the TA for the submission to be evaluated on your machine. Contact the TA if you have any questions.

GRADING CRITERIA

- **Late submissions will lose points as stated on the syllabus – 10% for each day that is late.**
- **A program that does not compile will result in a zero!**
- Functionality:
    - 10% - Program allows for all of the user input options correctly.
    - 30% - Program performs correctly while searching for the path.
    - 30% - Program outputs the correct paths from initial to final location when the “step-by-step” option is NOT selected.
    - 30% - Program outputs the correct paths from initial to final location when the “step-by-step” option is selected.

PLAGIARISM !

Your program must be your original work, as stated and described in the syllabus. If you are unsure about whether some open source code can be used, contact the TA.