Semester 2 2016
COMP3702/7702 ARTIFICIAL INTELLIGENCE
ASSIGNMENT 1: Navigation Problem

Note:

- This assignment consists of two parts: Programming and report.
- You can do this assignment in a group of at most 3 students. This means you can also do the assignment individually.
- For those who choose to work in a group:
  - All students in the group must be enrolled in the same course code, i.e., all COMP3702 students or all COMP7702 students.
  - Please register your group name in https://goo.gl/Ay98am before **11.59pm on Friday, 12 August 2016**. If you have not registered your group by the said time, you will need to work on the assignment individually.
  - All group members are expected to work in both programming and report. In the report, you are required to write the role of each team member.
- Submission Instruction:
  - Your program should compile using ant or make/cmake from command prompt, and generate an executable named a1-[courseCode]-[ID] that can be run from command prompt as:

    ```
    > a1-[courseCode]-[ID] environmentMapFileName queryFileName outputFileName
    ```

    Please replace courseCode with either 3702 or 7702, depending on which class you are enrolled in. If you work individually, please replace ID with your student ID. Otherwise, please replace ID with your group name.
  - The report should be in .pdf format and named a1-[courseCode]-[ID].pdf.
  - The report and all the source codes necessary to compile your program should be placed inside a folder named a1-[courseCode]-[ID]. Please submit only source codes (i.e., remove all object files).
  - The folder should be zipped under the same name, i.e., a1-[courseCode]-[ID].zip, and the zip file should be submitted via turnitin before **11.59pm on Friday, 26 August 2016**.

Ever wonder how the navigation app in your mobile phone works? Well, in this assignment , you have a chance to create your very own navigation program. Your task here is to develop a program that will accept as inputs:

- An environment map, represented as a graph. The graph is a weighted directed graph, where a vertex represents a location of interest. Throughout this assignment specification, we will use vertex and the state it represents interchangeably. A directed edge vv’ with label c means that there is a direct road to move from v to v’, and the cost of moving through the road is c.
- An initial and goal locations.

The program should output the shortest path to move from the initial to the goal locations.

To develop the navigation program, please:

1. Define the navigation agent.

2. Please use Uniform Cost and A* search algorithms to solve the problem faced by the navigation agent.

Input format

The input is two .txt files: One file contains the environment map, while another file contains the queries.

The environment map is represented as a graph. This graph is written in a matrix format: The matrix' size is nXn, where n is the number of vertices in the graph. Element [i, j] (row-i, col-j) of the matrix is the cost of moving from vertex-i to vertex-j. The cost of an edge is always a real number greater than 0.01. An element, say [i, j], with value 0 means there is no edge from vertex-i to vertex-j.
The file that contains this graph consists of n+1 lines. The first line contains only a single number: n. Each of the next n lines correspond to each row in the matrix, with line-k in the file corresponding to row-(k-1) in the matrix. The columns are separated by a single white space.

The queries file contains q+1 lines, where q is the number of queries. The first line contains only a single number: q. The rest of the files are the queries.
Each query is written in a single line, and consists of three components separated by a single white space. The first component is the type of algorithm to use. In this assignment, there will only be 2 types of algorithms: Uniform for Uniform Cost search and A* for A* search. The second component is the ID of the initial vertex, while the third component is the ID of the goal vertex.

Output format

The output file contains q lines, where q is the number of queries in the input file. Each solution (the shortest path) is written in a single line. The solution at line q must be the solution of the query at line q+1 in the input file. Each path should be written as a sequence of vertices separated by a dash ("-") sign. This sequence starts with the initial vertex and ends with the goal vertex.

Grading for the Programming Part (total points: 60/100)

For marking, we will use 4 different pairs of environment and queries files. Each queries file will contain 4 queries. Therefore, there will be a total of 16 queries. COMP3702 students can get a full mark by solving 12 of the 16 queries. However, COMP7702 students must solve all queries to get a full mark. Solving a query means finding the optimal path within the given time limit. The time limit may be different for different pairs of environment-queries files. The details of the grading scheme is as follows:

COMP3702:

- >= 1 & < 10: The program does not compile nor run.
- >= 10 & < 20: The program runs but fails to solve any query. The program fails to solve a query when it does not find a solution (i.e., a path with the least cost from the initial to the goal vertices) after more than 2X the given time limit.
- >= 20 & < 30: The program solves at least one of the queries within 1-2X the given time limit.

- >=30 & <= 60: The program solves at least one of the queries within the given time limit. Each query is worth 2.5 points.

COMP7702:

- >= 1 & < 5: The program does not compile nor run.
- >= 5 & < 10: The program runs but fails to solve any query. A program fails to solve a query when it does not find a solution after more than 2X the given time limit.
- >= 10 & < 20: The program solves at least one of the queries within 1-2X the given time limit.
- >=20 & <= 60: The program solves at least one of the queries within the given time limit. Each query is worth 2.5 points.

Report (total points: 40/100)

Your report must contain answers to the following questions:

1. [2.5 points] What is the formal definition of the navigation agent in this assignment?
2. [2.5 points] What type of agent is the navigation agent as described in question-1 (i.e., discrete / continuous, fully / partially observable, deterministic / non-deterimistic, static / dynamic)? Please explain your selection.
3. [7.5 points] What is the heuristic you use for A* search? Please explain why do you think it is a good heuristic?
4. [12.5 points] Please compare the performance (in terms of time and space) of Uniform Cost and A* search as the number of vertices in the graph increases. Please explain your findings. This explanation should include comparisons with the theoretical results.
5. [15 points] Suppose you are given 2 environment maps, say A and B. True or False that if the number of vertices in A is larger than in B, then given the same implementation of A* search, finding an optimal path in A will always take longer time than in B? Please explain your answer.

Please note that in each of the above question, when explanation is requested, the explanation part is 80% of the total points. For instance, in question 5, correct True or False answer will only give you 3 points, while a good explanation about your true or false answer could earn you up to 12 points.

Please also note that good explanation is NOT equal to long explanation!!!

oOo That's All, Folks oOo