

Programming Assignment 1

COMP 6651 – Winter 2019

Dr. Tiberiu Popa

For this exercise you are asked to solve the “*Traveling Salesman Problem*”: given a complete, undirected weighted graph of $N > 2$ vertices you have to find the minimum Hamiltonian cycle. A Hamiltonian cycle is a cycle that visits all vertices exactly once (with the exception of the first and last vertex). You can assume that the weights are positive, but you cannot assume that the weights obey the triangle inequality.

Specifications

The input is specified in a file whose name is the first argument of the program. The first line contains an integer M specifying how many datasets are in the file. The reminder of the file encodes the datasets. Each dataset consists of a definition of a graph as follows: It starts with two space separated positive integers V and E on the first line that indicates the number of vertices and edges respectively. The following lines specify the weights on the edges of the graph as following: each edge is specified on one line as 3 numbers: $u\ v\ w$. u is the index of the first vertex, v is the index of the second vertex and w is a positive integer weight of the edge (u,v) . Note that all edges of the graph appear in this list exactly once. (i.e. if we have an entry for (u,v) we will not have one for (v,u)).

Here is an example:

```
2
3 3
0 1 1
0 2 1
1 2 1
4 6
0 1 2
0 2 1
0 3 2
1 2 2
1 3 1
2 3 2
```

The output is a file whose name is the second argument of the program. Each line encodes the results of each test case. The algorithm should output the length of the minimal Hamiltonian cycle.

For example, the output corresponding to the input above is as follows:

```
3
6
```

What is given

No code is given for this programming assignment.

Submission

You have to implement your program in plain C/C++ in a file called *main.cpp* that has no dependency other than the standard C/C++ libraries available in a standard Linux system such as the one in the graduate labs. No external libraries are allowed even if they are available in our computing environment. The code should compile using the command ***g++ main.cpp*** in any machine in the graduate labs. You are not allowed to use the boost library. Any submission that include a boost header file receives a grade of 0.

You are not allowed to use any other flags to compile this homework.

The due-date is February 22nd, 5:00pm. Any late submission will receive a grade of 0. The only exceptions are medical and a note from a physician is required.

This exercise is individual.

Originality and Plagiarism

Some of the problems proposed in the course are “classical” in the algorithm design literature and, therefore, solutions may be available on-line. Please note that you are expected to do this problem individually and you are expected to produce an original solution. We run plagiarism tests and if your submission is red-flagged you are expected to explain it in detail to the instructor. Failure to comply with this request or to adequately explain your own code may result in filing a plagiarism case with the Dean of Graduate Studies.

Evaluation and Testing

This programming assignment will be evaluated automatically, but we do check if the code is original. You will receive 0 if the code does not compile. We run 3 test cases, you receive 1/10 marks if your code compiles, but it does not return the correct results on any of the test cases, 4/10 if it compiles and runs correctly on one of the test cases, etc.

You may be asked to explain your code to the teaching assistant or the instructor. This may lead to a decrease in your overall grade for this programming assignment if the code cannot be sufficiently explained.