

CPSC 350: Data Structures and Algorithms

Spring 2025

Programming Assignment 6: Spanning the Gamut

Due: See Canvas

Hints

Kruskal's algorithm might seem simple when we, as humans, carry it out. However, when transitioning to code, you might realize there are some issues we don't really think about when "drawing it out." Please check out [this slideshow](#) that illustrates the algorithm and its possible "complications".

Sample Input and Output

Input:

```
7
0.0 1.0 0.0 1.0 8.0 10.0 0.0
1.0 0.0 1.0 0.0 0.0 10.0 0.0
0.0 1.0 0.0 0.0 1.0 0.0 0.0
1.0 0.0 0.0 0.0 5.0 5.0 0.0
8.0 0.0 1.0 5.0 0.0 2.0 2.0
10.0 10.0 0.0 5.0 2.0 0.0 0.0
0.0 0.0 0.0 0.0 2.0 0.0 0.0
END
```

Output:

```
The MST Cost is: 8.0
0.0 1.0 0.0 1.0 0.0 0.0 0.0
1.0 0.0 1.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 1.0 0.0 0.0
1.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0 2.0 2.0
0.0 0.0 0.0 0.0 2.0 0.0 0.0
0.0 0.0 0.0 0.0 2.0 0.0 0.0
```

The Assignment

In this assignment, you will create a program that is capable of identifying a minimum spanning tree of an undirected, weighted graph using Kruskal's algorithm.

Your program will take as a command line argument the name of a file that contains the specification for an undirected, weighted graph, *G*. The file will have the following format:

- The first line will be an integer, *N*, that represents the number of nodes in the graph. You can assume the nodes are labeled with the values $[0, N-1]$.
- The next *N* lines will represent the rows of the adjacency matrix, with each line consisting of *N* weights, represented as doubles.
- The last line in the file will consist of the string "END."

You should process the file and create a weighted graph representation using the *WGraph* class we implemented together in class. Once the *WGraph* is created, your program should compute the MST for the graph using Kruskal's algorithm. This should be done by adding a method to *WGraph*, *computeMST()*, that works as follows:

1. Identifies an MST for the graph. Note that a given graph may contain several MSTs, but you only need to identify one.
2. Displays the cost of the MST (the sum of all the edges in the MST) to standard output
3. Displays the adjacency matrix representation of the MST (an NxN matrix where all edges are 0 except for the edges that make up the MST) to standard output

Note that you may create any additional classes you need to support your implementation (e.g. a priority queue to use for Kruskal's algorithm), but at a minimum, your solution should contain the following files:

- main.cpp
- WGraph.h
- WGraph.cpp

As part of your solution, it is useful to have a Set data structure that will allow you to keep track of nodes/edges you have already added to the MST and account for duplicates. To accomplish this, you may use the set class in the C++ standard template library (eg. `#include <set>`).

Rules of Engagement

- For this assignment, you must work individually.
- Develop using VSCode and make sure your code runs correctly with g++ using the course docker container.
- Feel free to use whatever textbooks or Internet sites you want to refresh your memory with C++ IO operations, just cite them in a README file turned in with your code. All code you write, of course, must be your own. In your README please be sure to include the g++ command for compiling your code.

Due Date

This assignment's due date is posted on Canvas. Submit all your commented code as a zip file to Canvas. The name of the zip file should be LastName_FirstInitial_A6.zip

Grading

Grades will be based on correctness, adherence to the guidelines, and code quality (including the presence of meaningful comments). An elegant, OO solution will receive much more credit than procedural spaghetti code. I assume you are familiar with the standard style guide for C++, which you should follow. (See the course page on Canvas for a C++ style guide and Coding Documentation Requirements.)

Code that does not follow the specification EXACTLY will receive an automatic 25% deduction. Code that does not compile will receive an automatic 50% deduction.

Readme & References

More info on Canvas!

- **README file:** All source code will be accompanied with a plain text README file. We encourage students to take advantage of markdown technology if they are inclined to do so. This file will contain:
 - The following identifying information:
 - Full name
 - Student ID
 - Chapman email
 - Course number and section
 - Assignment or exercise number
 - A list of all source files submitted for the assignment
 - A description of any known compile or runtime errors, code limitations, or deviations from the assignment specification (if applicable)
 - A list of all references used to complete the assignment, including peers (if applicable)
 - Instructions for running the assignment. (Typically applicable in advanced courses using build systems or third party libraries) for us it would be something like:
 - To compile: `g++ *.cpp -o A1.exe`
 - To run: `./A1.exe input.txt output.html`
- **In-code citations:** You are allowed to use small, isolated lines of code from external sources in your programming assignments as long as they are appropriately cited. Any time you are including code that you did not write yourself, it must be cited. This includes sources from StackOverflow and similar forums, other current or previous students, tutors, books, tutorials, etc. You should wrap the copied code in a comment denoting the start and end of said code, like this:

```
# code that is not copied (you wrote this)
someCode = 0;
print("hi I wrote this!")

""""BEGIN CODE FROM SOURCE: link/name of source""""
print("I did not write this");
""""END OF CODE FROM SOURCE: link/name of source""""

""""BEGIN CODE FROM CHAT GPT, PROMPT ASKED: how do you ...?
""""
print ("I did not write this");
""""END OF CODE FROM CHAT GPT""""

# back to code that you personally wrote
codeIWrote = 10;
```
- **Info in code files:** Additionally, all source files will start with a header comment containing the following items (one per line):

- Full name
- Student ID
- Chapman email
- Course number and section
- Assignment or exercise number