CS 5045
Homework Assignment 9

Given: November 7, 2016 **Due:** November 18, 2016

General directions. The point value of each problem is shown in [ ]. The completed assignment must be uploaded to Canvas as a single file by 5:00 PM EST on November 18, 2016. **No late homework will be accepted.**

---

[60] 1.

1 Introduction

This is a Python 3 programming assignment that develops a program to solve the minimum spanning tree (MST) problem using Prim's and Kruskal's algorithms. Your program, to be named `homework9.py`, uses the `argparse` module for parsing the command line and the `logging` module to log information about the progress of the program and about errors.

2 Files to Retrieve

From the Assignments page, retrieve the following files. It is suggested that you make a directory named `Homework9` and put all these files in that directory.

`template9.py`. This file is a starting point for your `homework9.py`. Retrieve it and rename it `homework9.py` before you start developing your solution.

`random_graph.py`. This contains a Python 3 function `random_weighted_graph` to generate a random graph with random edge weights.

`homework9.out`. This is a sample output file from running a complete `homework9.py`.

`homework9.log`. This is a sample log file from running a complete `homework9.py`.

3 Input Format

There is no input file. Each run of `homework9.py` generates its own random graph to use as input.

4 Output Format

The output file is a text file generated in the `main` function of `homework9.py`. See the sample output file to clarify the file format.

5 Your Command Line

You should be able to run your `homework9.py` program using a command line with this format:

```
homework9.py [-h] [-n number_nodes] [-p edge_probability]
        [-m maximum_weight] output_file
```

The `-n` command line option specifies the number of nodes in the random graph. The `-p` command line option specifies the probability of an edge between two nodes. The `-m` command line option specifies the maximum weight of an edge. See the implementation of `random_weighted_graph` to understand these three parameters.

6 Prim's and Kruskal's Algorithms

In `template9.py`, Prim's algorithm has already been implemented according to Listing 7-5, so you do not need to change that. Also, there are stubs for the three functions from Listing 7-4 needed to implement Kruskal's algorithm. You are to complete the implementation of Kruskal's algorithm according to Listing 7-4 and have the `kruskal` function return the tuple of the set of edges in the MST and the total weight of the MST.

As a check that you have it right, the total weight returned by `kruskal` should match the total weight returned by `prim`. Feel free to add additional Python instructions to check this.

7 Additional Questions

In comments (lines starting with `#`) at the end of your `homework9.py`, answer the following two questions.

1. Is it possible for `prim` and `kruskal` to return the same total weight but different edges? If not, why not? If so, when would it happen?

2. Is it possible for `prim` and `kruskal` to return different total weights and yet run correctly? Explain.

8 Hints

Ask questions early on Piazza if you do not understand the assignment.

Use the `logging` module to document successful processing as well as warnings and errors. See the sample log files for examples of thorough logging practice. See the documentation in the Python standard library for `argparse` and `logging` here:

`https://docs.python.org/3/library/index.html`

9 Submit

The completed program `homework9.py` must be uploaded to Canvas as a single file by 5:00 PM EST on November 18, 2016.