1. (7 points) Implement and compare Kruskel's and Prim's algorithms for minimum spanning tree (MST). Your algorithms take a weighted undirected graph as input and outputs a minimum spanning tree in the form of a set of edges and their total weight.

You can use python libraries to create random weighted graphs of different sizes and probability of connectivity using the so-called Erdos-Renyi model. The model called G(N,p) takes two paramaters N, the number of nodes, and p the probability of an edge being present. Set p = 2 ln N/N, which makes the graph connected with a very high probability. You can use a python package to generate random graphs in this model. For example , gnp_random_graph function in the networkx package seems appropriate.

[https://networkx.org/documentation/latest/reference/generators.html#module-networkx.generators.random_graphs (Links to an external site.)](https://networkx.org/documentation/latest/reference/generators.html#module-networkx.generators.random_graphs)

Assign each edge a random integer weight in the range of (1, 1000). Generate 10 graphs each of sizes 1100, 1200, ..., 2000 nodes. For each size graph, run both the algorithms and check that the weigths of the MSTs they find is the same. Report the average running times of the algorithms as a function of number of nodes in a plot/table. Comment on their relative performance. Is there a clear winner?

2. (1 point) You are driving on a long highway with gas stations at distances $d_1 < ... < d_n$ miles from the starting location *S*. Your car can run *M* miles with a full gas tank. You start with a full gas tank and want to reach the final location which is at d_n miles from *S*. How would you choose the gas stations to minimize the number of refueling stops? Justify why no other choice can make fewer stops.

3. (1 point) Let 'maximum spanning tree' be defined as a spanning tree with the maximum total weight. Define the *cut property* for maximum spanning tree as follows. Suppose *X* is a set of edges in a maximum spanning tree. Choose a set of vertices *S* such that no edges in *X* cross from nodes in *S* to nodes in *V-S*. Let *e* be the heaviest edge not in *X* that crosses from *S* to *V-S*. Show that *X union {e}* is a subset of a maximum spanning tree.

4. (1 point) A barber shop serves *n* customers in a queue. They have service times *t1,...tn*. Only one customer can be served at any time. The waiting time for any customer is the sum of the service times of all previous customers. How would you order the customers so that the total waiting time for all customers will be minimized? Carefully justify your answer.

Submission:

Label the source file 'hw6.py' and the report report6.pdf. Submit the report on canvas and the source at

https://teach.engr.oregonstate.edu/teach.php