Due: November 3, 4:00pm
(note: Third part of this lab will come out on November 3.)

This is a pair programming assignment. Continue working with your partner.

We will be using Python for this lab. To make things easy to manage and submit, you can develop most of your code in Jupyter notebooks. Your final submission may ask you to turn some of the code into a package, but it should be convenient for you to do all development for this and several other parts of the lab in Jupyter (or Google Colab).

In Part 1 of this lab you created a simple implementation of a disjoint sets Abstract Data Type (ADT) that supports four operations needed for maintaining the ADT and its use in the Kruskal's algorithm: `Init()`, `MakeSet()`, `FindSet()`, and `Union()`. In this part of the lab, you will complete the implementation of Kruskal's MST algorithm.

For this part of the lab, continue developing the `Lab5-MST.ipynb` notebook. For now, keep all developed code there. We will refactor it later.

**Task 1. Implementation of Kruskal's algorithm.** Using the disjoint sets ADT you developed in Part 1 of the lab, complete the implementation of Kruskal's MST algorithm. Name your function `kruskalMST`. It should take the following inputs:

- `G`: the adjacency matrix for the undirected edge-weighted graph whose MST you want to find.

The function shall return the Minimum Spanning Tree for the input graph. You can choose how to do this. There are two ways that are fairly straightforward:

- **Option 1.** Return the minimum spanning tree as an adjacency matrix, that only includes the edges that are part of the MST. Each edge shall include its weight.
- **Option 2.** Return the minimum spanning tree as a list of edges, where each edge is represented by three values: the two vertices that the edge connects, and the weight of the edge.
**Task 2. Testing of Kruskal's algorithm.** Implement a simple testing procedure that generates a random graph using the random graph generator functionality you developed in Part 1 of the lab, calls your implementation of Kruskal's algorithm, and prints the initial graph and the MST. It is left up to you how to print the graph and the MST information (it should be done in a somewhat readable way - at least for small graphs).

**Submission.** Place all this code into a python notebook named Lab5-MST.ipynb. At the end of Lab 5 you will submit this notebook as one of your deliverables, but as you will add code to it, there is no need to submit anything just yet.