Introduction:

Graphs are a data structure that represent connections among the node items which are called vertices and are connected with edges. The edges can have a weight on them depending the type of graph it is. There will be two types of graph implementations we will be constructing in this lab. The first one is called the Kruskal’s algorithm and it is a minimum-spanning-tree **algorithm** which finds an edge of the least possible weight that connects any two trees in the forest making it what it is called a “greedy” algorithm. The second one is called topological sort and it is supposed to determine the order in which every vertex is supposed to be stored in a graph.

Proposed solution design and implementation:

Truthfully graphs were my favorite part of the semester and getting to implement a homework’s algorithm was so much fun! The Kruskal’s graph on the homework had weighted edges while the topological one did not but it had directed edges. In the code you can trace how these would be drawn out. Both implementations were able to be completed with out zybooks resource.

There is a file for each algorithm implementation and an execute file for the hard coded graphs for each. In the Kruskal’s algorithm it needed to be modified to insert edges and to preview them in the console. Then in the topological sort algorithm did what happened in the above algorithm and additionally keeps in mind what edge has been visited.

A close up of text on a black background

Description automatically generated

A screenshot of a cell phone

Description automatically generatedExperiment:

Conclusion: Tracing these algorithms manually for me

had to more fun than seeing it on a console. However

it was interesting to implement it.