CSCI 3330 Project 3

Graph algorithms

Graph theory and applications have been broadly employed in modeling various kinds of applications. In this course this semester, we have introduced a number of algorithms involving graphs. In our programming exercises, you have implemented a list of graph algorithms. In this project, you are asked to apply these algorithms to answer some related questions involving theory and applications.

Project objectives:

Through completing this project, you should able to

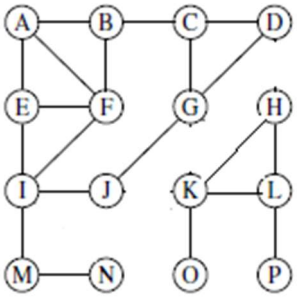
- Represent and process graph objects computationally;
- Distinguish common properties and differences of some graph algorithms;
- Solve applications modeled with graphs; and
- Enhance oral and writing communication skills through group work and report writing.

Project description:

This project is built up on the programming assignments of Chapters 3, 4 and 5. Although you may apply the sample programs introduced in class, you are encouraged (but not required) to apply an available graph ADT, for instance NetworkX in Python, to complete this project.

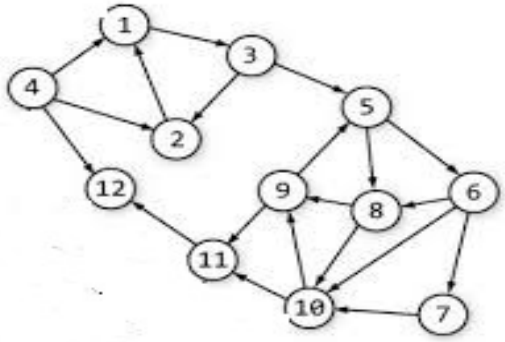
Here are the required tasks:

1. You are asked to represent the undirected graph below in a computer. Then, apply it as a sample to answer the following questions computationally:



   a) Starting from any vertex, can DFS and BFS find all connected components of an undirected graph?
   b) Can both BFS and DFS determine if there is a path between two given nodes?
   c) There is a path between two vertices A and B. If started from A, do DFS and BFS always find exactly the same path?

2. A connected digraph can be decomposed to its strongly connected components as a 'meta graph'. Implement the digraph below as a graph object, and then answer the following questions computationally.

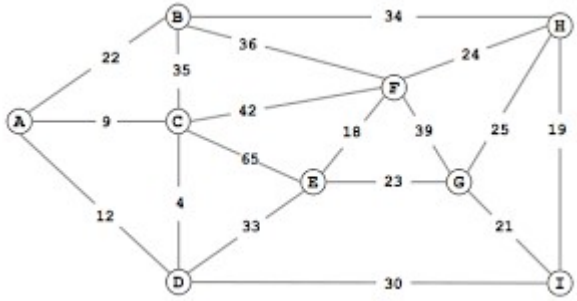


   a) Use an application to find the strongly connected components of the digraph;

   b) Draw the digraph as a 'meta graph' of its strongly connected components in your project report (Note: you don't have to draw it with a program. Draw it manually is acceptable); and then

   c) Represent the 'meta graph' as a DAG and linearize it in its topological order.

3. Provided is a weighted undirected graph. Implement it as a graph object and then computationally answer the following questions:



   a) Write an application that applies Dijkstra's algorithm to produce the shortest path tree for a weighted graph with a given starting node. Test and verify your program with the given graph starting with node A;

   b) Write a program that produces a minimum spanning tree for a connected weighted graph. Test your program with the given graph above;

   c) Are a shortest path tree and a minimum spanning tree usually the same?

   d) If the graph has an edge with a negative weight, can you apply Dijkstra's algorithm to find a shortest path tree?

What to submit: You should submit your source code together with a brief report. In the report you need to answer the questions after each problem with computational outputs (from your program or hand calculation) to support your answer. You MUST specify the contribution of each group member and properly credit the sources of any references (including code).

How it will be graded: I will test run your program, and then check the completeness. The grading is mostly determined by the correctness of the answers of the questions and your computational outputs.