

Assignment 4

Graphs

Purpose

The goal of this lab is to become familiar with the graph data structure and its applications.

Background

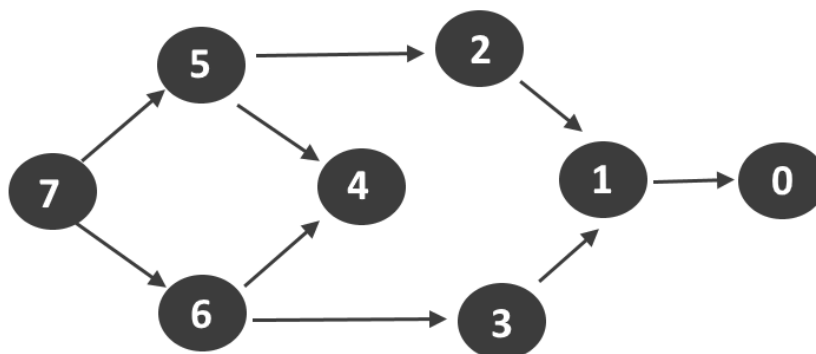
Graph: is a non-empty finite set V of elements called vertices together with a possibly empty set E of pairs of vertices called edges. $G(V, E)$

1. Topological Sort

1.1 Introduction

Topological sorting for Directed Acyclic Graph is a linear ordering of vertices such that for every directed edge uv , vertex u comes before v in the ordering. We need to print a vertex before its adjacent vertices.

Topological Sorting only works for Directed Acyclic Graphs. And it's not unique.



Topological Sort : 7 6 5 4 3 2 1 0

Check visualization for more understanding

<https://www.cs.usfca.edu/~galles/visualization/TopoSortIndegree.html>

1.2 Requirements

Given: Directed Acyclic Graph.

Goal: To find a list of nodes in topologically sorted order.

You're required to implement the above procedure.

2. Minimum Spanning Tree

2.1 Introduction

Theorem: Every connected graph has a spanning tree.

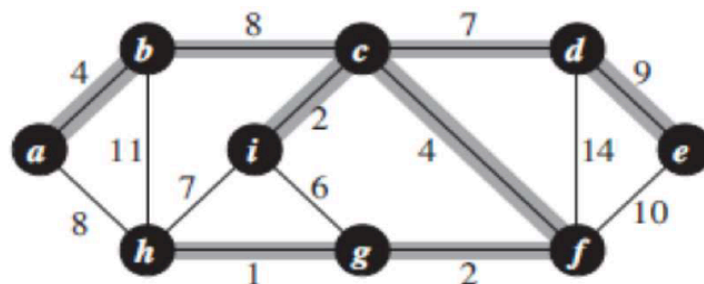
Weighted Graphs: A weighted graph is a graph, in which each edge has a weight (some real number).

Weight of a Graph: The sum of the weights of all edges.

A Minimum Spanning Tree in an undirected connected weighted graph is a spanning tree of minimum weight (among all spanning trees).

The minimum spanning tree may not be unique. However, if the weights of all the edges are pairwise distinct, it is indeed unique.

MST Problem: Given a connected weighted undirected graph G , design an algorithm that outputs a minimum spanning tree (MST) of G .



Prof Dr. Amr El Masry
Dr. Mohamed Kholeif

Prim's Algorithm

The Prim's algorithm makes a nature choice of the cut in each iteration – it grows a single tree and adds a light edge in each iteration.

Check visualization for more understanding:

<https://www.cs.usfca.edu/~galles/visualization/Prim.html>

2.2 Requirements

Given: Weighted undirected graph, single source s .

Goal: Find Edges (MST) using Prim's algorithm starting from s to include all vertex.

You're required to implement the above procedures, pseudo code for the above procedures are explained in details in tutorials

3. Notes

- Implement your algorithms using (Python, Java, or C/C++).
- You should work in groups **of 3 members**.
- Discussion will have higher weight than implementation, so you should understand your implementation well to get discussion marks.
- Late submissions are not allowed unless there is a valid documented excuse.