

Data Structures and Algorithms

LAB #9 – Graphs

Fall 2018

Objectives

By the end of this lab, the student should be able to:

- Implement graph using adjacency list and adjacency matrix representation
- Convert from one representation to another
- Implement common graph algorithms (e.g. Dijkstra's, Prim's,... etc)

Code Examples

The give code has three examples:

- 1- Adjacency List → AdjList VC project
Shows how to implement a graph using adjacency list representation
- 2- Adjacency Matrix → AdjMatrix VC project
Shows how to implement a graph using adjacency Matrix representation
- 3- Dijkstra's Algorithm → Dijkstra VC project
Shows the implementation of Dijkstra algorithm
As an example, the code finds the shortest path from node zero to all other nodes in graph of figure1-a

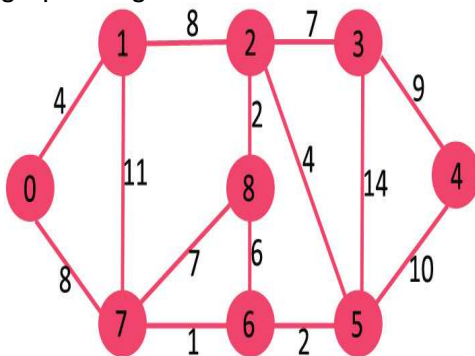


Figure 1-a
Graph example

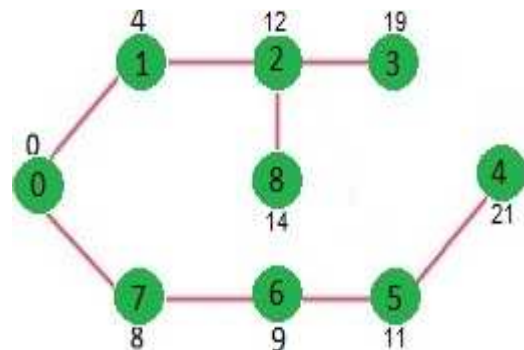


Figure 1-b
Shortest path from node 0 to other nodes

Practice Exercises

1. Write a function **GraphAdjList GraphAdjMatrix::ConvertToAdjList()**
The function is a member in class **GraphAdjMatrix** and converts from adjacency matrix representation to adjacency list representation and returns an object of class **GraphAdjList**
2. Similarly, write **GraphAdjMatrix GraphAdjList::ConvertToAdjMatrix()**
3. Write a member function **GraphAdjMatrix::BreadthFirst** that traverses the graph in breadth-first order
4. Implement the following graph algorithms:
 - a. Topological sorting
 - b. Kruskal's minimum spanning tree