Tutorial 11

Introduction

This tutorial focuses on Graph algorithms like graph traversal and finding a topological order in a graph.
Example 01 demonstrates how to represent a graph G= {V, E} using adjacency matrix, and the implementation of BFS algorithm to traverse the graph.
Example 02 shows a representation of a graph G= {V, E} using adjacency list, and the implementation of DFS algorithm.
In the exercise section, students are asked to:

- Implement an algorithm to find all connected components in a graph.
- Find a topological order in a graph.

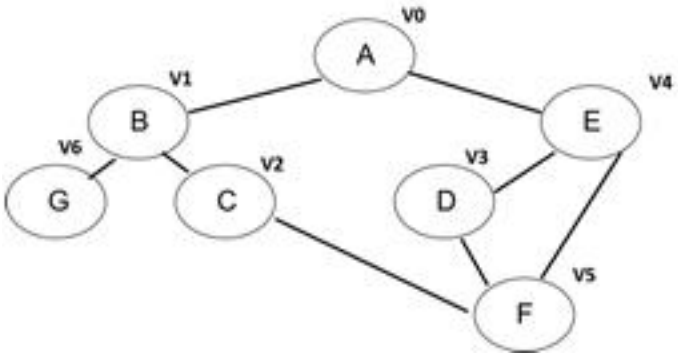
Examples

1. Example 01 – BFS implementation

A graph G={V,E} is represented by using an adjacency matrix of the size |V|x|V| (or NxN, where N is the number of vertices)

$$a[i][j] = \begin{cases} 1 & \text{if } (vi, vj) \in E \\ 0 & \text{if } (vi, vj) \notin E \end{cases}$$

In this graph G, each vertex has a label that is simply a character assigned to the vertex. An example of G and its adjacency matrix is the following:

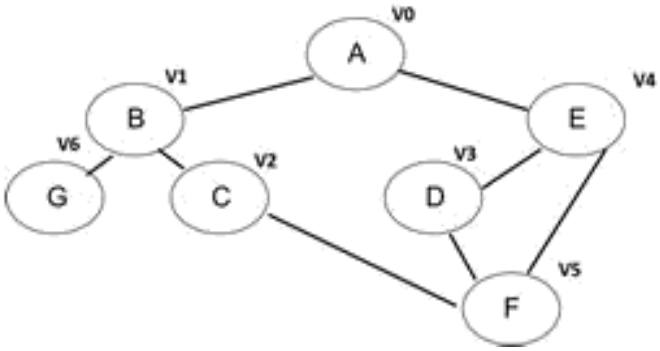


	0	1	2	3	4	5	6
0	0	1	0	0	1	0	0
1	1	0	1	0	0	0	1
2	0	1	0	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	0	1	0
5	0	0	1	1	1	0	0
6	0	1	0	0	0	0	0

The example demonstrates the implementation of BFS and the use of the algorithm for graph traversal. Class GVertex is designed to abstract a vertex of G. Class GraphBFS implements the algorithm. Please refer to class ***GVertex and GraphBFS*** in the Tutorial Example Code.

2. Example 02 – DFS implementation

In the adjacency list representation, a graph G= {V,E} contains a list of N vertices, each vertex maintains a list of its own adjacency vertices. Similar to example 01, a character label is assigned to each graph vertex. An example of G and its adjacency matrix is the following:



Vertex	Adjacency list
V0	{V4 → V1}
V1	{V6 → V2 → V0}
V2	{V5 → V1}
V3	{V5 → V4}
V4	{V5 → V3 → V0}
V5	{V4 →V3 →V2}
V6	{V1}

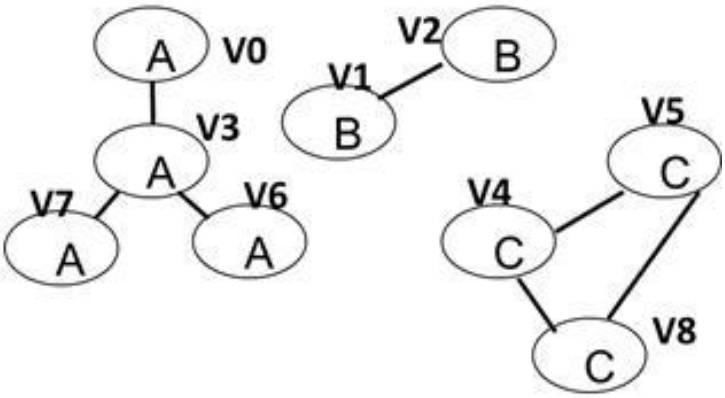
The example demonstrates the implementation of DFS and the use of the algorithm for graph traversal. Class GALVertex is designed to abstract a vertex of G. In order to represent the adjacency list the singly linked list of integer node is used. Class GraphBFS implements the algorithm. Please refer to class ***SLNode, SLList, GALVertex and GraphDFS*** in the Tutorial Example Code.

Exercises

1. Exercise 1

In a graph G= (V, E), we want to assign a character label to each vertex of G using the following rule:

- Label starts from ‘A’ to ‘Z’.
- Every vertices in the same connected component of G will be assigned using the same label. The example below shows a graph with assigned labels:

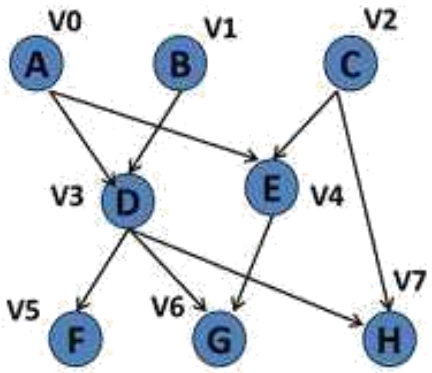


	V0	V1	V2	V3	V4	V5	V6	V7	V8
V0	0	0	0	1	0	0	0	0	0
V1	0	0	1	0	0	0	0	0	0
V2	0	1	0	0	0	0	0	0	0
V3	1	0	0	0	0	0	1	1	0
V4	0	0	0	0	0	1	0	0	1
V5	0	0	0	0	1	0	0	0	1
V6	0	0	0	1	0	0	0	0	0
V7	0	0	0	1	0	0	0	0	0
V8	0	0	0	0	1	1	0	0	0

Given a graph G represented by an adjacency matrix of size NxN, please design and implement an algorithm to assign labels to graph's vertices.

2. Exercise 2

Given a directed graph G= {V, E} that is represented using an adjacency matrix. Implement an algorithm to find a topological order of this graph.



	V0	V1	V2	V3	V4	V5	V6	V7
V0	0	0	0	1	1	0	0	0
V1	0	0	0	1	0	0	0	0
V2	0	0	0	0	1	0	0	1
V3	0	0	0	0	0	1	1	1
V4	0	0	0	0	0	0	1	0
V5	0	0	0	0	0	0	0	0
V6	0	0	0	0	0	0	0	0
V7	0	0	0	0	0	0	0	0