

CS2040C Semester 1 2018/19
Data Structures and Algorithms

Tutorial 09 - Graph Representation and Graph Traversal

For Week 11 (Week Starting 29 October 2018)

Document was last modified on: October 28, 2018

1 Introduction and Objective

This tutorial marks the beginning of our discussion on graphs. Firstly, this tutorial will discuss the various graph data structures needed to store and represent graphs. Then, some examples of real life scenarios that can be modelled as a graph will be discussed. The tutorial will then end off with a quick introduction/recap to Depth-First-Search or Breadth-First-Search.

Basic Stuff About Graph DSes

Q1). Tutor will draw just one random small graph on the **whiteboard** first and ask students to store that graph in either Adjacency Matrix (AM)/Adjacency List (AL)/Edge List (EL) data structure on the **whiteboard**. Then, the tutorial group can compare the answers by drawing the same small graph on <https://visualgo.net/en/graphds>.

This is also an open ended exercise. Tutor will actually do two more rounds of reusable exercises on top of the published task above:

The conversion challenge: Ask the student to **convert** the graph data structure that they chose his/herself into another one that we have discussed in class (AM/AL/EL) **after** erasing the original graph drawing (so that they can't rely on the original picture unless they have photographic memory).

Various Type of Graphs

Q2). Draw a Directed Acyclic Graph (DAG) with V vertices and $V \times (V - 1)/2$ directed edges.

Easy with experience but can be non trivial for beginners.

For DAG: Just draw a line containing V vertices labeled from 0 to $V - 1$. Then draw directed edge from a vertex i to a vertex j for all pairs of (i, j) where $i < j$, that's all.

Follow up question: Are you sure that $V \times (V - 1)/2$ edges is the maximum possible for a DAG with V vertices? Can we add more edge(s)? Answer: No, adding any more edge will create a cycle.

Q3). Draw a Bipartite Graph with V vertices (assume that V is even) and $V^2/4$ undirected edges.

This question has been integrated in VisuAlgo Online Quiz :).

For Bipartite: Put $V/2$ vertices on the left and the other $V/2$ vertices on the right and link them all. This is the configuration that yield the most edges.

Graph Modeling Exercise Part 1

Q4).]In tutorial, the tutor will randomly imagine a real life scenario (that can be modeled as a graph problem) and will ask random student to model that scenario into a graph. Students have to describe what the set of vertices represent, what the set of edges represent, are the graphs weighted/directed/connected?, what are the graph (for now, limit to graph traversal) problem being asked?, etc...

This part is left to the tutor. Suggested examples (please limit the graph problems to Graph Traversal type only, don't ask MST (outside of CS2040/C syllabus)/SSSP/advanced graph problems yet): Facebook/social network friendship graph, want to measure if two user accounts are connected directly (or via some chain of intermediate friends); I-Know-You (bidirectional) graph of students in the tutorial group, want to find biggest group of students who know each other directly (or indirectly) in the tutorial group; Singapore MRT network, want to find stations that are not more than k -stations away from a given stations, etc...

DFS/BFS Review

Q5). The tutor will end the tutorial by quickly reviewing DFS and BFS graph traversal algorithms using <https://visualgo.net/en/dfsbfbs> starting from a randomly drawn graph. We will discuss harder applications of these two graph traversal algorithms in the next tutorial.

This part is left to the tutor. As this is nearing the end of the course, please help ensure that CS2040/C students understand the basic ideas of DFS and BFS graph traversal algorithms before diving into harder stuffs that we can do with those two basic graph traversal algorithms.

Problem Set 4

That's the end of Tut09. You can ask your tutor about PS4 and Practical Exam if you encounter any difficulties with it.