Learning Guide Unit 3

Site: University of the People
Course: CS 3304-01 Analysis of Algorithms - AY2024-T5
Book: Learning Guide Unit 3

Printed by: Mejbaul Mubin
Date: Saturday, 29 June 2024, 6:43 AM

Description

Learning Guide Unit 3

Table of contents

Overview

Unit 3: Graphs (Part 1)

Topics:

- Graph terminology and representation
- Graph data structures
- Graph traversals
- Greedy Algorithms

Learning Objectives:

By the end of this Unit, students will be able to:

1. Describe directed, undirected, labeled graphs and subgraphs
2. Be able to describe vertices and edges
3. Be able to implement graphs as data structures
4. Understand and be able to implement graph traversals including:
    - Depth-first search
    - Breadth-first Search
    - Topological sort
5. Understand the basic characteristics and operation of greedy algorithms

---

Tasks:

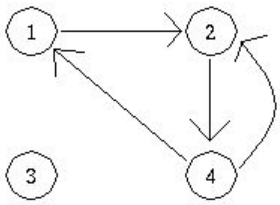
- Peer assess Unit 2 Programming Assignment
- Read the Learning Guide and Reading Assignments
- Participate in the Discussion Assignment (post, comment, and rate in the Discussion Forum)
- Complete and submit the Programming Assignment
- Make entries to the Learning Journal
- Take the Self-Quiz

Introduction

Unit three will extend our understanding of graphs by looking at a class of problems that are solved using graphs and finding the shortest path through a graph. We can think of many such problems that exist in real life. For those who have experience with project management techniques you will recognize project planning as a shortest path through a graph type of problem. Another important example is the need to plot a path through the internet to get from your computer and your browser to a destination web server such as the one that we use for the University of the People web site or Google. When you consider that the internet is projected to have 15 billion nodes by 2015 and over 50 billion by 2020 the efficient traversal of the internet (which of course is a graph structure) is vital.

Often as we begin to study algorithms we might be tempted to ask why it is so important that we understand theoretical concepts such as the 'growth' in execution time or execution complexity and the example of the internet is an easy one to grasp. The scale of the internet and a variety of other modern computing problems has made it increasingly important for us as computer scientists to understand algorithm complexity and to be able to evaluate how a particular algorithm will scale as the size of n grows large … large such as n=50,000,000,000 as is the case in the internet in just a few short years.

If you will recall from CS3303 an important operation in a graph is a transversal. A traversal is simply an ordered way to visit each node or vertex within the graph in a specific order. We typically represent a graph using circles connected with lines or arrows. The circles are called vertices and the lines that connect them are edges.



In graph terminology, when the lines have no arrow head on them the graph is said to be undirected. When the lines have arrow heads it indicates the required direction of travel along the edge and the graph is said to be directed. A directed graph is also referred to as a digraph.

When labels (either numbers or letters) appear in the vertices, the graph is said to be labeled.

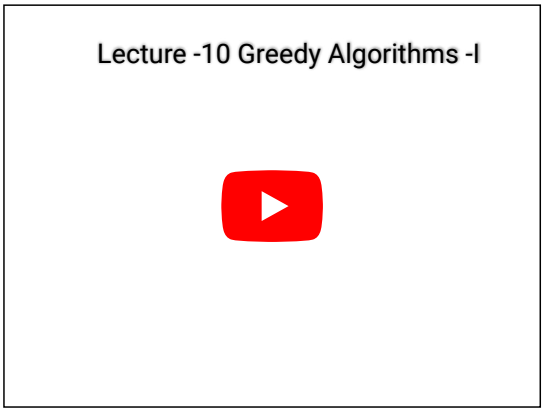
In this unit we want to understand two key items. First is to become familiar with the terminality of graphs and second is to understand the rules and processes associated with different types of graphtraversals.

Greedy Algorithms

Greedy algorithms look for simple, easy-to-implement solutions to complex, multi-step problems by deciding which next step will provide the most obvious benefit. Such algorithms are called greedy because while the optimal solution to each smaller instance will provide an immediate output, the algorithm doesn't consider the larger problem as a whole. Once a decision has been made, it is never reconsidered.

The advantage to using a greedy algorithm is that solutions to smaller instances of the problem can be straightforward and easy to understand. The disadvantage is that it is entirely possible that the most optimal short-term solutions may lead to the worst long-term outcome.

If you have access to a computer with sufficient network bandwidth, then you might want to review the following lecture on greedy algorithms in the series on Design & Analysis of Algorithms by Prof.Abhiram Ranade, Prof.Sunder Vishwanathan, Department of Computer Science Engineering,IIT Bombay.



Reading Assignment

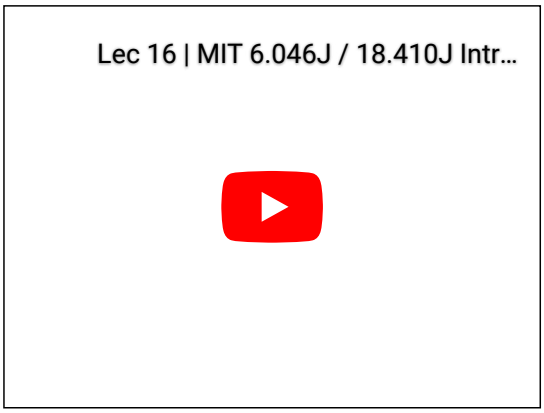
Topic 1: Graphs as data structures
Chapter 11 Graphs, Sections 11.1 – 11.3 in A Practical Introduction to Data Structures and Algorithm Analysis by Clifford A. Shaffer.
Optionally review Chapter 3 Decomposition of Graphs in Algorithms by S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani available at http://www.cs.berkeley.edu/~vazirani/algorithms/chap3.pdf

Topic 2: Graph Traversals
Chapter 11 Graphs, Sections 11.1 – 11.3 in A Practical Introduction to Data Structures and Algorithm Analysis by Clifford A. Shaffer.
Optionally review Chapter 4 Paths in Graphs in Algorithms by S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani available at http://www.cs.berkeley.edu/~vazirani/algorithms/chap4.pdf

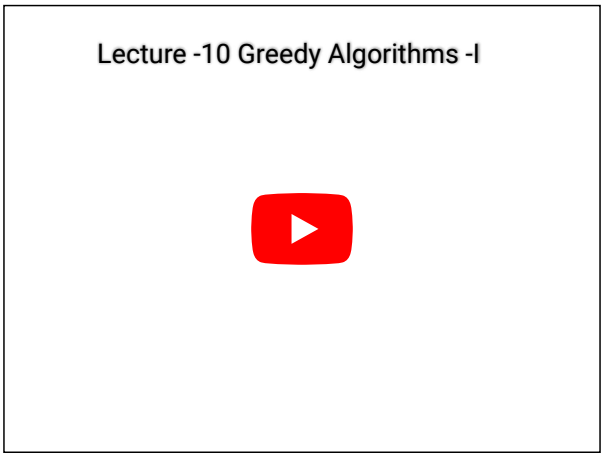
Topic 3: Greedy Algorithms
Read the lecture 14 notes from Design and Analysis of Algorithms available from: http://www.cse.ust.hk/~dekai/271/notes/L14/L14.pdf
Read Chapter 5 Greedy Algorithms in Algorithms by S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani available at http://www.cs.berkeley.edu/~vazirani/algorithms/chap5.pdf

Supplemental Materials
The following are video lectures that are available via YouTube and other sources that are related to the topics in the unit and can be used as a supplemental resource for students looking for more details or to be introduced to the same topic from another source. The use of these resources is not required and is entirely optional.

Greedy Algorithms (and Graphs) By Charles E. Leiserson – MIT



Lecture on Greedy Algorithms by Prof. Abhiram Ranade and Prof. Sunder Vishwanathan, Department of Computer Science Engineering, IIT Bombay.



Unit 3 Optional Video Lectures

The following video lectures are optional resources that have been made available to students who can take advantage of them. These lectures are strictly optional resources. All of the information in these lectures is available in other learning resources within the course. These lectures are provided for those students who have sufficient network bandwidth and technology capabilities to take advantage of video content. These lectures cannot be used instead of the required assigned resources and there is no information that is not contained in the assigned resources. These lectures simply present some of the information in a different format.

- Unit 3 Lecture 1: Graphs as data structures
- Unit 3 Lecture 2: Graph Traversals
- Unit 3 Lecture 3: Greedy Algorithms

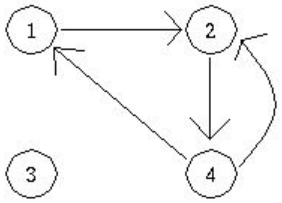
Discussion Assignment

Describe and compare both the depth-first and breadth-first search as a graph traversal. As part of your discussion, describe under what conditions or which problem each is best utilized to solve. Finally, your discussion must incorporate a description of how such traversals are implemented as greedy algorithms.

Include one or two examples to explain your thought process to show what is occurring and how the methodology works. Use APA citations and references for any sources used.

Programming Assignment

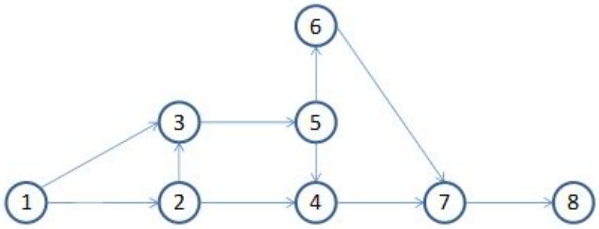
Although this assignment doesn't technically require any 'development' in terms of Java programming or programming in any other language, it is development in terms of developing a solution and a data structure. Consider the following picture representation of a graph:



This graph can be implemented using the following data structure.

V = {1, 2, 3, 4}
E = { (1, 2), (2, 4), (4, 2) (4, 1)}

Create a data structure to represent the graph detailed in the following picture:



As part of your assignment you must also determine (and include as part of the discussion in your assignment) if the graph is:

- Acyclic or not
- Directed or undirected
- Connected or not
- Simple or not

Please explain all of your answers.

Learning Journal

The Learning Journal is a tool for self-reflection on the learning process. In addition to completing directed tasks, you should use the Learning Journal to document your activities, record problems you may have encountered and to draft answers for Discussion Forums and Assignments. The Learning Journal should be updated regularly (on a weekly basis), as the learning journals will be assessed by your instructor as part of your Final Grade.

Your learning journal entry must be a reflective statement that considers the following questions:

- Describe what you did. This does not mean that you copy and paste from what you have posted or the assignments you have prepared. You need to describe what you did and how you did it.
- Describe your reactions to what you did
- Describe any feedback you received or any specific interactions you had. Discuss how they were helpful
- Describe your feelings and attitudes
- Describe what you learned

Another set of questions to consider in your learning journal statement include:

- What surprised me or caused me to wonder?
- What happened that felt particularly challenging? Why was it challenging to me?
- What skills and knowledge do I recognize that I am gaining?
- What am I realizing about myself as a learner?
- In what ways am I able to apply the ideas and concepts gained to my own experience?

Your Learning Journal should be a minimum of 500 words

Self-Quiz

The Self-Quiz gives you an opportunity to self-assess your knowledge of what you have learned so far.

The results of the Self-Quiz do not count towards your final grade, but the quiz is an important part of the University's learning process and it is expected that you will take it to ensure understanding of the materials presented. Reviewing and analyzing your results will help you perform better on future Graded Quizzes and the Final Exam.

Please access the Self-Quiz on the main course homepage; it will be listed inside the Unit.

Checklist

- [ ] Peer assess Unit 2 Programming Assignment
- [ ] Read the Learning Guide and Reading Assignments
- [ ] Participate in the Discussion Assignment (post, comment, and rate in the Discussion Forum)
- [ ] Complete and submit the Programming Assignment
- [ ] Make entries to the Learning Journal
- [ ] Take the Self-Quiz