Learning Guide Unit 4

Site:	University of the People	Printed by:	Mejbaul Mubin
Course:	CS 3304-01 Analysis of Algorithms - AY2024-T5	Date:	Saturday, 29 June 2024, 6:44 AM
Book:	Learning Guide Unit 4		

Description

Learning Guide Unit 4

Table of contents

Overview

Unit 4: Graphs (Part 2)

Topics:

- Shortest Path Problems
- Minimum Cost Spanning Tree
- Prim’s Algorithm
- Kruskal’s Algorithm

Learning Objectives:

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

1. Describe Shorted Path Problems
2. Define Minimum-cost spanning tree and approaches to implement it
3. Understand and apply algorithms to solve shortest path problems
    - Prim’s algorithm
    - Kruskal’s algorithm
4. Articulate the characteristics of ‘greedy’ algorithms

Tasks:

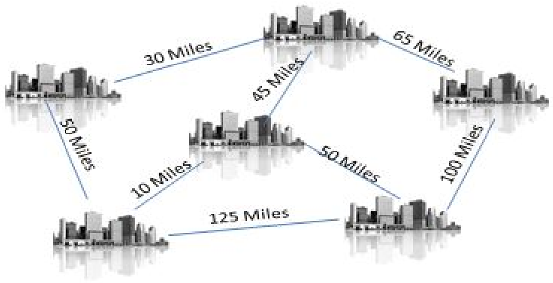
- Peer assess Unit 3 Programming Assignment
- Read the Learning Guide and Reading Assignments
- Participate in the Discussion Assignment (post, comment, and rate in the Discussion Forum)
- Complete and submit the Programming Assignment
- Make entries to the Learning Journal
- Take the Graded Quiz

Introduction

Unit four will continue our exploration of graphs. In unit 3 we learned about the graph as a structure and some of the applications of graphs. In this unit we will continue our examination of graphs and learn how efficient algorithms to traverse a connected graph can be developed. Further we will be looking a special class of traversal which is designed to identify the shortest path through a graph.

Greedy Algorithms

Many algorithms can be formulated as a finite series of guesses. Consider the Travelling Salesman Problem. Imagine a traveling salesman who has to visit each of a number of cities on his route. Each city has different distances from other cities as illustrated in the following diagram.



The question is what is the most efficient or shortest path that will enable the salesman to visit each city? How can we determine the shortest path? We try (guess) each possible tour in turn and determine its distance (cost). When we have tried them all, we know which one is the optimum (least cost) one. However, we must try them **all** before we can be certain that we know which is the optimum one, leading to an ***O(n!)*** algorithm.

Intuitive strategies, such as building up the salesman's tour by adding the city which is closest to the current city, can readily be shown to produce sub-optimal tours. As another example, an experienced chess player will not take an opponent's pawn with his queen - because that move produced the maximal gain, the capture of a piece - if his opponent is guarding that pawn with another pawn. In such games, you must look at all the moves ahead to ensure that the one you choose is in fact the optimal one. All chess players know that short-sighted strategies are good recipes for disaster!

There is a class of algorithms, the *greedy algorithms*, in which we can find a solution by using only knowledge available at the time the next choice (or guess) must be made. The problem of finding the Minimum Spanning Tree (path that the salesman must take) is a good example of this kind of algorithm.

Prim's algorithm

Prim's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected weighted undirected graph. The way that Prim's algorithm works is that it selects a single vertex (node of the graph) and then evaluates the cost of each edge that would form a path to another node. The algorithm will select the path with lowest cost that does not form a cycle within the graph. A cycle within a graph is a situation where an edge within the graph must be traversed multiple times.

Kruskal's algorithm

Kruskal's algorithm is an algorithm that will identify a MINIUMUM spanning tree. This menas that for a weighted tree, the algorithm will find a path along the edges that will allow every vertex or node in the tree to be visited while minimizing the cost of the path in terms of the total cost of the edges. This description may be a bit cryptic so we can put this into the form of an example. Suppose that we need to find a path through the nodes of our local LAN (another term for this might be a 'route') and we want to minimize the amount of time that it requires to accomplish this. The time to send a packet of information between any two nodes in our network will have a cost that is expressed in time. Network engineers will refer to this time cost at 'latency'. To ensure that we can move data efficiently through our network we need a way to discover the most efficient path through our network. An efficient path is one that minimizes the latency.

Kruskal's algorithm is a good example of a greedy algorithm as it builds the structure one step at a time. The algorithm will begin at one vertex in the graph and then evaluate potential path that can be taken to the next vertex and selects the one that has the smallest cost.

Reading Assignment

Topic 1: Shortest Path and Minimum Spanning Tree
Chapter 11 Graphs, Sections 11.1 – 11.3 in A Practical Introduction to Data Structures and Algorithm Analysis by Clifford A. Shaffer.
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

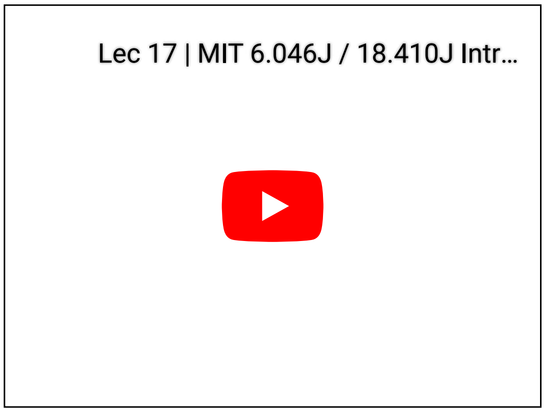
Topic 2: Prim's Algorithm
Chapter 11 Graphs, Sections 11.1 – 11.3 in A Practical Introduction to Data Structures and Algorithm Analysis by Clifford A. Shaffer.
Read Lecture 7: Minimum Spanning Trees and Prim's Algorithm available at http://www.cse.ust.hk/~dekai/271/notes/L07/L07.pdf

Topic 3: Kruskal's Algorithm
Chapter 11 Graphs, Sections 11.1 – 11.3 in A Practical Introduction to Data Structures and Algorithm Analysis by Clifford A. Shaffer.
Read Lecture 8: Lecture 8: Kruskal's MST Algorithm available at http://www.cse.ust.hk/~dekai/271/notes/L08/L08.pdf

Read the Wikipedia article on Kruskal's algorithm paying special attention to the pictoral representation of how Kruskal's algorithm creates a minimum spanning tree from a collection of graphs that are assembled together. Available at
http://en.wikipedia.org/wiki/Kruskal's_algorithm

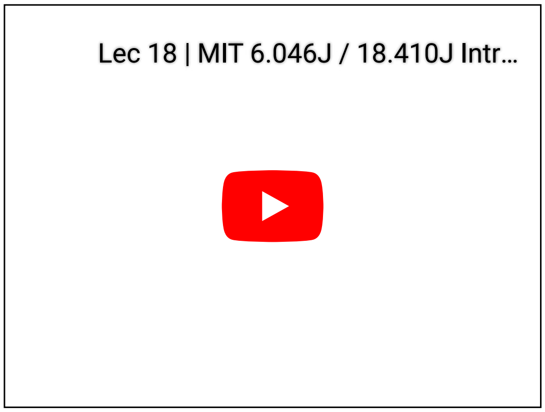
Supplemental Materials
The following are video lectures that are available via YouTube and other sources that are related to the topics in the unit and can be used as a supplemental resource for students looking for more details or to be introduced to the same topic from another source. The use of these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

Shortest Paths I, By Erik Demaine – MIT



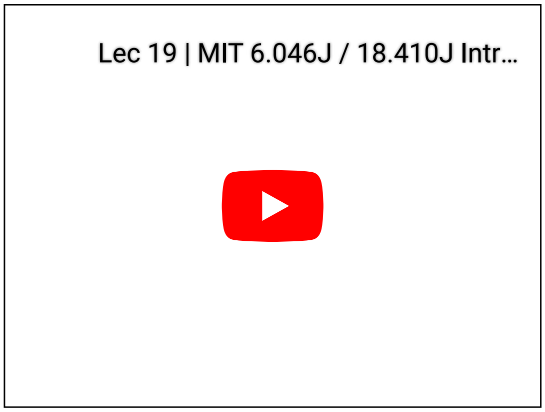
Lec 17 | MIT 6.046J / 18.410J Intr...

Shortest Paths II, By Erik Demaine – MIT



Lec 18 | MIT 6.046J / 18.410J Intr...

Shortest Paths III, By Erik Demaine – MIT



Lec 19 | MIT 6.046J / 18.410J Intr…

Unit 4 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 4 Lecture 1: Minimum path and Spanning Tree
- Unit 4 Lecture 2: Prim's Algorithm
- Unit 4 Lecture 3: Kruskal's Algorithm

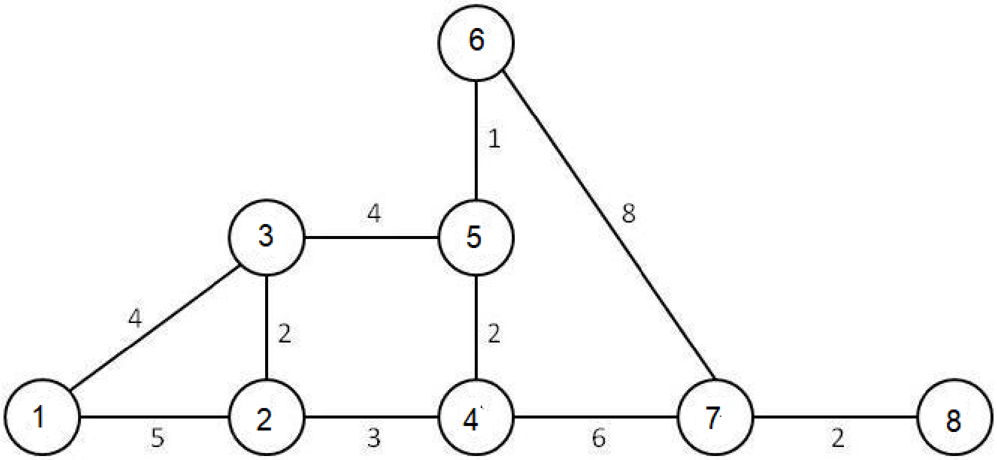
Discussion Assignment

In Chapter 11 of the Shaffer text, we are introduced to Kruskal’s Algorithm for determining a Minimum Spanning Tree within a graph. This algorithm is described as a greedy algorithm. Describe in your own words how Kruskal’s algorithm works and be sure to articulate in your description what makes Kruskal’s algorithm ‘greedy’.

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

Develop an implementation of Prim's algorithms that determines the MST (Minimum Spanning Tree) of the graph from the Unit 2 assignment that we developed the data structure for.



For this assignment, develop an implementation using Java in the Cloud9 environment (or your own Java IDE) that first implements the graph in a data structure and then provides the algorithm that can determine the Minimum spanning tree within this graph in terms of cost. The cost will be the sum of the lengths of the edges that must be traversed. The cost of each edge is represented by the number on the edge. For example the cost of edge 1,3 is 4 and the cost of edge 6,7 is 8. Your algorithm must output the total cost of spanning the tree as determined by your implementation of Prim's algorithm. The algorithm must produce output which is the total cost of the path.

Assessment
You will have ONE WEEK to complete this assignment. It will be due the end of this unit. Your assignment will be assessed (graded) by your peers. You should post this assignment, the results, and other requirements such as the asymptotic analysis in one of the following formats:

- Directly cut-and-pasted into the text box for the posting.
- As a document in either RTF or Word 97/2003 format.

Mark each rubric item using the scale provided. Some scales will be simple yes/no responses. Others will be a scale of 1-4 where 4 indicates that all of the elements of the rubric item are present and 1 indicates that NONE of the elements were present. For example in the rubric item "Was a java implementation of a minimum spanning tree provided", then if no code was submitted at all then this item should be 1. If an algorithm was posted (not necessarily working as that would be another rubric item) that attempts to implement a minimum spanning tree traversal, then this item should be 4.

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.

Graded Quiz

The Graded Quiz will test your knowledge of all the materials learned thus far. The results of the quiz will count towards your final grade.

Please access the Graded Quiz on the main course homepage; it will be listed inside the Unit. After you click on it, the quiz’s introduction will inform you of any time or attempt limits in place.

Good luck!

Checklist

Peer assess Unit 3 Programming Assignment

Read the Learning Guide and Reading Assignment

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

Take the Graded Quiz.