#### INTRODUCING GRAPH THEORY IN HIGH SCHOOL EDUCATION

by

#### Eva Brittina Riddick

A Thesis submitted to the Graduate Faculty of Elizabeth City State University in partial fulfillment of the requirements for the Degree of Master of Science in Mathematics.

Elizabeth City, North Carolina

May

 $20\,19$ 

APPROVED BY

Julian A.D. Allagan, Ph.D. Committee Chair Kenneth L. Jones, Ph.D. Committee Member

James Ogaja, Ph.D. Committee Member Colby Hunter, Ph.D. Committee member ©Copyright 2019 Eva Brittina Riddick All Right Reserved

#### ABSTRACT

This thesis presents three fundamental research topics commonly discussed in Graph Theory, an important area of Discrete Mathematics. Our objective is to propose activities and problems that will strengthen the math skills of North Carolina high school students as they meet a required competency goal in Discrete Mathematics. For this reason, we discuss several topics such as Spanning trees, Minimum Spanning trees, Euler and Hamilton Graphs, and Vertex Coloring. Activities for practice and several applied problems are proposed with detailed solutions to achieve several learning objectives.

#### DEDICATION

In dedicating this thesis, I must first give honor and glory to Almighty God for creating me and empowering me to complete this part of my life's journey. His mercies being new everyday, Lamentations 3:23 in the Bible, and His perfect will strengthened me so that I could endure and complete this program in the midst of the losses of my dear loved ones and my personal trials.

Secondly, I want to dedicate this thesis to my paternal grand-parents, Harold and Eva Mullen, my maternal grandparents, Rev. Walter and Brittina Brothers, my mother, Hannah E. Melson, my two dads Percy Mullen and Franklin Melson Sr., my husband Charles, our children Raymond, C.J., and Shania, our grandchildren, Jamir, Emiyjah, Nizier, and Zac'Quiere, and to all of my siblings, my sisters by the Blood of Jesus, Bessie, Anissa, Eld. Dorene Carruth Norris & the ladies of Serious Worship, and every friend who encouraged me to keep pressing forward and reminded me that " faith without work is dead," James 2:17.

#### ACKNOWLEDGEMENT

My sincere and utmost gratitude goes to Dr. Julian Apelete Allagan who worked tirelessly with me on this project. Thank you so much for your compassion and understanding when I was going through such a hard time in my life. Thank you so much for always quickly responding to my emails for clarity of what I needed to do throughout the development of this thesis. Thank you, Dr. Allagan, for your respect, your professionalism, your wisdom, and your guidance this semester. I also want to extend my appreciation to Dr. Kenneth Jones, the current chairman of the Mathematics Department, Dr. Farrah Ward, the former chairman of the Mathematics Department, all of my class instructors in the master's program, Dr. Paula Viltz, and to all of the librarians who helped me to locate the information that pertained to my topics or to find a file or document that I had misplaced. Thank you Mrs. Claudia Davis and Ms. Chiquita Mitchell for providing transportation to and from school when I had no car. I must also thank my high school Algebra II teacher and current mayor of Elizabeth City, Mrs. Bettie Parker, for being my inspiration to become a math teacher and for encouraging me to enter the master's program. Last, but not least, I must acknowledge Dr. Sohindar Sachdev and Dr. Louise N. Sutton for teaching me from the level I was and raising me up, and not from the level they were, in my undergraduate math classes. God bless you all!!!

## Contents

1	Introduction 1						
	1.1	NC High School Math	2				
	1.2	Mathematical Modeling	3				
	1.3	Basic Graph Theory	4				
	1.4	Basic definitions	5				
	1.5	Special Graphs	6				
		1.5.1 Subgraphs	6				
		1.5.2 Spanning subgraphs	6				
		1.5.3 Paths	7				
		1.5.4 Cycles	7				
		1.5.5 Trees	7				
		1.5.6 Complete graphs	8				
		1.5.7 Bipartite graphs	8				
	1.6	Activity: Basic graph notion	8				
	1.7	Answers					
2	Sna	Spanning Troos 19					
2	2 1	Definition 1	12				
	2.1 Dominion						
	2.2	2.2.1 Incidence Matrix	-2 19				
		2.2.1 Incluence Matrix	.2 12				
		2.2.2 Adjacency Matrix	.ð				
		2.2.3 Degree Matrix	.3				
	2.3	Laplacian Matrix	13				
		2.3.1 Counting spanning trees	13				

	2.4	Minimum spanning trees	16
		2.4.1 Kruskal's algorithm	16
		2.4.2 Prim's algorithm	18
	2.5	Activity: Spanning trees of a graph	19
	2.6	Answers	20
3	Eul	erian and Hamiltonian Graphs	29
U	Lur		20
	3.1	Eulerian Graph and Cycle	29
	3.2	Hamiltonian Graph and Cycle	30
	3.3	Activity	31
	3.4	Answers	33
4	Ver	tex Coloring	36
	4.1	Activity	38
	4.2	Answers	39
5	Cor	clusion and Future Research	43

# List of Figures

1.1	Mathematical Modeling Process	3
1.2	Konigsberg city and its corresponding graph model $\ldots$	4
1.3	Example of a simple graph on 6 vertices	5
1.4	A Wheel graph $G$ and its subgraph $H$	6
1.5	A Wheel graph $G$ and its spanning subgraph $H$	6
1.6	A Path on 3 vertices	7
1.7	A Cycle on 5 vertices	7
1.8	A tree on 6 vertices	7
1.9	A family of four complete graphs	8
1.10	A complete bipartite graph with parts sizes 3 and 2 $\ldots$ .	8
1.11	Intersection graph	11
2.1	A Wheel graph with two of its spanning trees	12
2.2	A graph $G$	15
2.3	The three spanning trees of $G$	16
2.4	A weighted communication graph $G$	17
2.5	A weighted graph $G$	18
2.6	Some simple graphs $P, Q$ , and $R$	19
2.7	Some graphs $S, T$ , and $U$	19
2.8	Some weighted simple graphs $V, W$ , and $X$	20
2.9	A Weighted Graph Showing Monthly Lease Costs for Lines in a Com-	
	puter Network	21
2.10	A Weighted Graph of some towns in Nevada	22
2.11	A minimum spanning tree of Graph X from Figure 2.8 $\ldots$ $\ldots$	27

3.1	Some simple graphs named $G_1, G_2, G_3$ (from left to right)	30
3.2	Some simple graphs named $G_4, G_5, G_6$ (from left to right)	31
3.3	Some simple graphs named $A, B, C$ and $D$ (from top-left to bottom-	
	right)	32
3.4	A floor plan	32
3.5	Flights cost between each pair of cities	33
3.6	Floor plan graph	34
4.1	A 3-colorable planar graph.	36
4.2	Two simple graphs $G$ and $H$	37
4.3	Two properly colored simple graphs $G$ and $H$	38
4.4	Committee Graph showing committees that share a member in common.	41
4.5	Courses Graph with no scheduling conflict	41
4.6	Radio stations located within 150 miles or less	42

# List of Tables

1.1	Test Specification Weights for the Discrete Mathematics NCFE 2003	
	Standard Course of Study [4]	3
3.1	List of different routes for the sales person starting from Detroit $\ .$ .	35
4.1	Distance between pairs of 6 six radio stations	38

### Chapter 1 Introduction

Mathematics is one of the core subjects of the world's educational system. In the state of North Carolina high school students are required to successfully achieve twenty-two credits to graduate from high school, four of which must be in math. The courses are Math I, Math II, Math III, with the fourth math course to be aligned with the student's post high school plans. According to North Carolina Standard Course Of Study (NCSCOS) [3], the choices for the  $4^{th}$  Level Math Courses are Advance Functions and Modeling, Integrated Math IV, Pre-Calculus, and Discrete Math. This thesis focuses on the fourth level math course–Discrete Mathematics by introducing related topics and activities that are designed to strengthen students core skills in this area of Math. These topics, although simple, are often discussed both at the college and graduate school levels, in a specific course called Graph Theory. As such, we introduce the reader or the student to some basic concepts of graph theory at the same time as we deliver core math skills for the state high school final exam. We also note several recent researchers also tried to discuss similar graph theory topics that are deemed accessible to high school students in other states. See [6], for instance.

The thesis is briefly outlined as follows: In this introductory chapter (Chapter 1) we discuss the compentency goal and objectives as we introduce the reader to some basic definitions and notions of graph theory. In Chapter 2, we introduce the first topic, Spanning trees, which meets two learning objectives. In chapter 3, we introduce the second topic, Euler and Hamiltonian Graphs, as we meet another learning objective. In Chapter 4, we introduce the third topic, Vertex coloring, as we reinforce the same learning objective covered by the topic discussed in Chapter 3. Related activities, and application problems are given with their solutions at the end of each Chapter. We close the thesis with a possible research direction and

some recommendations in Chapter 5. Several acronyms we used are defined in the Appendix 5.

## 1.1 NC High School Math

Discrete Mathematics introduce students to the mathematics of networks, social choice and decision making. The course extends students' applications of matrix arithmetic and sometimes probability. Applications and modeling are central to this course of study. Modeling in general is one of the required NC HS Standards for Mathematical Practice. For example, in NC Math I and NC Math II modeling makes an appearance in F-LE. 1-5, A-CED.1-3, and S-ID.7-9. However, the only place that an entire unit focused on mathematical modeling appears in the NC HS Collaborative Instructional Framework is in NC Math III–Modeling with Geometry unit . These acronyms are others can be found in the Appendix 5.

Moreover, seniors enrolled in Discrete Mathematics in the state of North Carolina are required to take a final exam issued by the state. According to the North Carolina Assessment Specifications (NCAS) of the 2018-2019 North Carolina Final Exam (NCFE) of Discrete Mathematics, the purpose of the assessment is to measure students' academic progress on the North Carolina Standard Course of Study adopted by the North Carolina State Board of Education (NCSBE) in June 2003. The NCSBE policy TEST-016 directs schools to use the results from all course-specific NCFEs as a minimum of twenty percent (20%) of the student's final course grade. A school district can assign a higher percentage to the final exam, if they so choose. The Discrete Mathematics course covers 3 main goals or competencies and each goal has 3-7 objectives. Goal 1 focuses on Graph Theory, Goal 2 focuses on Probability and Statistics while Goal 3 focuses on Recurssions and Series. This thesis focuses only on **Competency Goal 1** and its objectives which account for about (30%) of discrete math course standard requirement as stated in [4]. We state this goal and list its 3 main objectives; we note that they are not equally weighted as shown in Table1.1.

COMPETENCY GOAL: The learner will use matrices and graphs to

#### model relationships and solve problems.

1.01.a—Use matrices to model and solve problems.

1.01.b—Write and evaluate matrix expressions to solve problems.

1.02—Use graph theory to model relationships and solve problems.

Table 1.1: Test Specification Weights for the Discrete Mathematics NCFE 2003 Standard Course of Study [4]

Standard	
1.01	$\approx 18\%$
1.02	$\approx 12\%$
2.01	$\approx 15\%$
2.02	$\approx 24\%$
2.03	$\approx 21\%$
3.01	$\approx 9\%$
Total	100%

Standard Percent of Total Score Points

## 1.2 Mathematical Modeling

Mathematical modeling refers to the use of mathematical approaches to understand and make decisions about real world phenomena (See[2]). The modeling process starts and ends in a real-world context. The diagram below represents a flow chart of the process.



Figure 1.1: Mathematical Modeling Process

The first phase requires that students identify variables, essential features of the context, and make assumptions to narrow the messiness of the problem. In the next phase, formulation or representation, students mathematize the problem by creating and selecting geometric, algebraic, or statistical representations that describe the relationships between the variables. What they obtain is the **mathematical model**.

For our purpose, we will be using **graphs** to represent our mathematical model of a real world problem. Graphs are discrete structures consisting of vertices and edges that connect these vertices. There are different kinds of graphs, depending on whether edges have directions, whether multiple edges can connect the same pair of vertices, and whether loops are allowed. For the purpose of this thesis loops will not be allowed. In almost every conceivable discipline problems can be solved using graph models. Some examples are graphs to model acquaintanceships between people, collaboration between researchers, telephone calls between telephone numbers, and links between websites. Graphs can also be used to model roadmaps and the assignment of jobs to employees of an organization.

# 1.3 Basic Graph Theory

Graph theory began in 1736 when Leonard Euler published a paper that contained the solution to the 7 bridges of Konigsberg (see Figure 1.2 left) problem. Is it possible to take a walk around town crossing each bridge exactly once and wind up at your starting point? A graph (vertices and links) is used to model or represent the Konigsberg problem (see Figure 1.2 right). The answer to this problem is "no". The reason is discussed in Chapter 3. To help provide a solution to this problem, Euler used a drawing or a model that we call *graphs*, that reduces the problem down to its important elements, thus avoiding unnecessary details. We begin by introducing the basic information about graphs.



Figure 1.2: Konigsberg city and its corresponding graph model

## **1.4** Basic definitions

A simple graph G = (V, E) consists of V = V(G), a nonempty set of objects called vertices (or nodes) and E = E(G), a set of an unordered pair of distinct vertices called edges.



Figure 1.3: Example of a simple graph on 6 vertices

See Figure 1.3, for example. Vertices, say u and v that share an endpoint are said to be **adjacent**; u is also said to be a neighbor of v and vice-versa the edge denoted by uv is said to be incident to the vertices u and v. The **order** of the graph G is the size of its vertex set which we denote by |V| and the size of the edge set, denoted by |E|, is called **size** of the graph G. The degree of vertex, v denoted by deg(v), is the number of edges incident to v; that is the size of its neighbor. A vertex of degree 0 is said to be **isolated** while a vertex of degree 1 is called a **leaf**. The **minimum degree** of G, denoted by  $\delta(G)$ , is its smallest vertex degree, and the **maximum degree** of G denoted by  $\Delta(G)$  is the largest degree among its vertices. A vertex u is said to be connected to a vertex v, in a graph G, if there exists a sequence of edges (or path) from u to v in G. A graph G is **connected** if there is a path that connects every two of its vertices. There are other types of graphs such as multigraphs (when multiple edges are allowed between vertices), pseudographs (when a vertex is allowed to be connected to itself, as in a loop) and directed graphs (when each edge is given an orientation, using an arrow). However, our Thesis will focus only on simple graphs, as previously defined.

## 1.5 Special Graphs

Here, we present some common special graphs.

### 1.5.1 Subgraphs

Given a graph G with vertex set V(G) and edge set E(G), we call a graph H a **subgraph** of G if the vertex set  $V(H) \subseteq V(G)$  and the edge set  $E(H) \subseteq E(G)$ ; H is obtained from G by deleting edges (including incident vertices) and/or vertices from G. Below is an example of a graph G called *Wheel*, and its subgraph H (in red) called *cycle*; it is obvious that H is obtained from G by deleting the middle vetex and its incident edges.



Figure 1.4: A Wheel graph G and its subgraph H

### 1.5.2 Spanning subgraphs

Suppose H is a subgraph of G. If V(H) = V(G) and  $E(H) \subseteq E(G)$ , then H is said to **span** G. For example, from the Wheel graph G, we can obtain the spanning graph H (in red), called a *Star* graph. See Figure 1.5 for an example.



Figure 1.5: A Wheel graph G and its spanning subgraph H

## 1.5.3 Paths

A path of length n, denoted by  $P_n$ , is a graph that has exactly 2 leaves and every other vertex is of degree 2. Below is an example of a  $P_3$ .



Figure 1.6: A Path on 3 vertices

## 1.5.4 Cycles

A **cycle** on *n* vertices, denoted by  $C_n$  is a graph with exactly one closed path. Here is a  $C_5$ , a cycle on 5 vertices.



Figure 1.7: A Cycle on 5 vertices

### 1.5.5 Trees

A tree also known as an **acyclic** graph on n vertices, denoted by  $T_n$  is a graph with no cycle. Here is a  $T_6$ , a tree on 6 vertices.



Figure 1.8: A tree on 6 vertices

### 1.5.6 Complete graphs

A complete graph also known as cliques on n vertices, denoted by  $K_n$  is a graph where every pair of vertices are adjacent. Below is a family of complete graphs,  $K_2$ ,  $K_3$ ,  $K_4$ , and  $K_5$  (from left to right).



Figure 1.9: A family of four complete graphs

### 1.5.7 Bipartite graphs

A simple graph G = (V, E) is called **bipartite** if its vertex set be divided into two disjoint groups, with edges connecting vertices from one group to the other; no edge connects vertices within the same group. We note that when each vertex from one group is connected to each vertex from the group, the resulting bipartite graph is said to complete; we write  $K_{m,n}$  where m, n, are the sizes of the two groups. Below is *complete bipartite* graph  $K_{3,2}$ .



Figure 1.10: A complete bipartite graph with parts sizes 3 and 2

## **1.6** Activity: Basic graph notion

#### I. Practice:

Given the graph in Figure 1.3, answer the following questions.

- (a) List out the elements of each of these two sets: V(G) and E(G).
- (b) List the degrees of each vertex of G.
- (c) What is the value of  $\Delta(G)$ ?
- (d) What is the value of  $\delta(G)$ ?
- (e) True or False:
  - (i) The sum of the degrees of G is twice the size of G.
  - (ii) The sum of the degrees of G is twice the order of G.
- (f) Give an example of a connected subgraph of G consisting of four vertices.
- (g) Give an example of a connected subgraph of G that contains no cycle.
- (h) Give an example of a connected spanning subgraph of G of size 5 that contains no cycle.
- (i) Is the graph G connected?
- (j) Draw the following connected graphs: K<sub>6</sub>, K<sub>3,3</sub>, and T<sub>7</sub> with a maximum degree
   5.

#### **II.** Application:

The intersection graph of a collection of sets  $A_1, A_2, \ldots, A_n$  is the graph that has a vertex for each of these sets and has an edge connecting the vertices representing two sets if these sets have a nonempty intersection. Construct/draw the intersection graph of these collections of sets (label the vertices appropriately).  $A_1 = \{0, 2, 4, 6, 8\}, A_2 = \{0, 1, 2, 3, 4\}, A_3 = \{1, 3, 5, 7, 9\}, A_4 = \{5, 6, 7, 8, 9\}, A_5 = \{0, 1, 8, 9\}.$ 

**Note:** Although we are proposing only 1 applied problem here, several application questions in the upcoming chapters give the student additional modeling problems which require graph drawings.

## 1.7 Answers

#### I. Practice:

- (a) Vertex set:  $V(G) = \{a, b, c, d, e, f\}$ ; Edge set:  $E(G) = \{ad, af, bd, be, cd, ce\}$ .
- (b) Vertices degrees: deg(a) = 2, deg(b) = 2, deg(c) = 2, deg(d) = 4, deg(e) = 2, deg(f) = 2.
- (c) The maximum degree is  $\Delta(G) = 4$ .
- (d) The minimum degree is  $\delta(G) = 2$ .
- (e) True or False:
  - (i) True-each edge contributes to two vertices, hence two degrees. So the sum of its degrees is 2+2+2+2+2+2+4 = 14 = 2 × 7, since G has 7 edges, which is its size.
  - (ii) False-The order (number of vertices) of the graph is 6, while the sum of its degrees is 2+2+2+2+2+2+4 = 14.
- (f) The cycle with vertices b, a, c, e from G.
- (g) A Star on 5 whose vertices are d, f, a, b, c.
- (h) Example of a connected spanning subgraph of G that contains no cycle.



- (i) Yes, since no vertex is isolated.
- (j) Below are the graphs (from left to right)  $K_6$ , a complete graph on 6 vertices,  $K_{3,3}$ , a complete bipartite graph and  $T_7$ , a tree with a maximum degree 5.



#### **II.** Application:

The vertices are  $A_1, A_2, \ldots, A_5$ . Two vertices are connected, if their corresponding sets share an element. For instance,  $A_1$  and  $A_2$  have at least one element (0, 2, 4) in common, thus they will be connected while,  $A_1$  and  $A_3$  don't have any member in common; they will not be connected. See Figure 1.11, for a possible drawing.



Figure 1.11: Intersection graph

### Chapter 2 Spanning Trees

This chapter is designed to prepare the student for NCSBE competency goal 1– objectives 1.0.1.a and 1.0.1.b.

## 2.1 Definition

A spanning tree of a connected graph G is a tree that is a subgraph of G and contains every vertex of G. Below is a graph G and two of its subgraphs  $H_1$  and  $H_2$ .



Figure 2.1: A Wheel graph with two of its spanning trees

# 2.2 Number of spanning trees subgraphs

There are several ways to represent graphs using matrices. We list here four kinds, although only the last three will be important for us to compute the number of spanning trees in a graph.

### 2.2.1 Incidence Matrix

Let G = (V, E) be a simple. Suppose that  $v_1, v_2, \ldots, v_n$  are the vertices and  $e_1, e_2, \ldots, e_m$ are the edges of G. Then the incidence matrix with respect to this ordering of V and E is the  $n \times m$  matrix  $M = [m_{ij}]$ , where

$$m_{i,j} = \begin{cases} 1 & \text{when the edge } e_j \text{ is incident with } v_i \\ 0 & \text{otherwise.} \end{cases}$$

## 2.2.2 Adjacency Matrix

Another common way to represent graphs is through adjacency matrices. It is a square matrix with entries 0 or 1; 1 when a pair of vertices are adjacent and 0 otherwise.

### 2.2.3 Degree Matrix

A degree matrix is a diagonal matrix with vertex degrees on the diagonals, and 0's off-diagonal.

## 2.3 Laplacian Matrix

Given a simple graph G with n vertices, its Laplacian matrix is the square matrix L, which is computed by

$$L = D - A,$$

where D is the degree matrix and A is the adjacency matrix of the graph.

#### 2.3.1 Counting spanning trees

Let A be a square matrix. The **minor** of the element  $a_{ij}$  is denoted by  $M_{ij}$  and is the determinant of the matrix that remains after deleting row *i* and column *j* of A. The **cofactor** of  $a_{ij}$  denoted by  $C_{ij}$  is given by

$$C_{ij} = (-1)^{i+j} M_{ij}.$$

We assume the reader is capable of computing the determinant of a  $2 \times 2$  matrix

and so we begin by reminding the reader about how to compute the determinant of a  $3 \times 3$  matrix using a well-known formula:

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = a \cdot \det \begin{pmatrix} e & f \\ h & i \end{pmatrix} - b \cdot \det \begin{pmatrix} d & f \\ g & i \end{pmatrix} + c \cdot \det \begin{pmatrix} d & e \\ g & h \end{pmatrix}$$

**Example 2.3.1.** Find det(A), the determinant of the given matrix A.

$$A = \begin{pmatrix} 2 & -1 & 2 \\ -1 & 2 & 1 \\ 4 & 3 & 1 \end{pmatrix}.$$

Using the previous formula, we obtain  $\begin{pmatrix} 2 & 1 & 2 \end{pmatrix}$ 

$$\det \begin{pmatrix} 2 & -1 & 2 \\ -1 & 2 & 1 \\ 4 & 3 & 1 \end{pmatrix} = 2 \cdot \det \begin{pmatrix} 2 & 1 \\ 3 & 1 \end{pmatrix} - (-1) \det \begin{pmatrix} -1 & 1 \\ 4 & 1 \end{pmatrix} + 2 \cdot \det \begin{pmatrix} -1 & 2 \\ 4 & 3 \end{pmatrix}$$
$$= 2(-1) - (-1)(-5) + 2(-11) = -29$$

**Example 2.3.2.** Determine the minors and the cofactors of the elements  $a_{11}$  and  $a_{32}$  of the following matrix A.

$$A = \begin{pmatrix} 1 & 0 & 3 \\ 4 & -1 & 2 \\ 0 & -2 & 1 \end{pmatrix}$$

Solution: Applying the above definitions, we get the following:

- (i.) Minor of  $a_{11}$ : after deleting row 1 and column 1 of A, we obtain the resulting matrix  $\begin{pmatrix} -1 & 2 \\ -2 & 1 \end{pmatrix}$  whose determinant we compute. Thus,  $M_{11} = det \begin{pmatrix} -1 & 2 \\ -2 & 1 \end{pmatrix} = (-1 \times 1) (2 \times (-2)) = 3.$
- (ii.) The cofactor of  $a_{11} = (-1)^{1+1}M_{11} = (-1)^2M_{11} = (1)(3) = 3$ . Similarly, we have

(iii.) Minor of  $a_{32}$ : after deleting row 3 and column 2 of A, we obtain the resulting matrix  $\begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix}$  whose determinant we compute. Thus,  $M_{11} = det \begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix} = (1 \times 2) - (3 \times 4) = -10.$ 

(iv.) The cofactor of  $a_{32} = (-1)^{3+2}M_{32} = (-1)^5M_{32} = (-1)(-10) = 10.$ 

The number of distinct spanning trees in a graph can be computed in polynomial time through **Kirchhoff's (matrix-tree) theorem**[1] which states that such number is equal to any cofactor of its Laplacian matrix.

**Example 2.3.3.** Determine the number of spanning trees of the graph G below.



Figure 2.2: A graph G

#### Solution:

(i.) First, we compute the Adjacency Matrix A and the Degree Matrix D of G.

(ii.) Next, we determine the Laplacian matrix of G by computing L = D - A.

$$L = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

(iii.) Finally we use any of the entries to compute the cofactor of L. Using, say,  $a_{11}$ , we have:

(a.) the minor of 
$$a_{11}$$
 is  $M_{11} = \det \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = 3.$   
(b.) the cofactor of  $a_{11} = (-1)^{1+1} M_{11} = (-1)^2 M_{11} = (1)(3) = 3.$ 

Here are the 3 spanning trees of the graph G in Figure 2.3.



Figure 2.3: The three spanning trees of G.

**Remark 2.3.1.** It is obvious that for any element  $a_{i,j}$ ,  $|M_{ij}| = |C_{ij}|$ , we recommend finding the number of spanning trees by simply computing  $|M_{ij}|$ .

## 2.4 Minimum spanning trees

A graph whose edges are labeled with numbers (known as **weights**) is called a **weighed graph**. See Figure 2.4, for instance of a graph G where the vertices represent 12 nodes and the edges are weighted according to the distance between these nodes.

A minimum-weight spanning tree, or simply a **minimum spanning tree**, is a spanning tree for which the sum of the weights of all the edges is as small as possible. There are few known algorithms for finding the minimum spanning tree of a given graph. Here, we discuss two commonly used.

#### 2.4.1 Kruskal's algorithm

**Description**:



Figure 2.4: A weighted communication graph G.

In Kruskal's algorithm, the edges of a connected weighted graph are considered one-by-one in an increasing order of weights (Step 1). At each stage the edge being considered is added (or highlighted) to what will become the minimum spanning tree, as long as that this addition does not create a circuit or cycle; in which case we discard that edge (Step 2). After n - 1 edges are added we stop, as we have a tree which is considered a minimum spanning tree for the graph.

**Example 2.4.1.** Use Kruskal's algorithm to find the minimum spanning tree of the graph G in Figure 2.4.

Let's consider the graph in Figure 2.4 and apply Kruskal's Algorithm. We first note the weights of each edge. cd = 1, bf = 1, kl = 1, ab = 2, cg = 2, fj = 2, bc = 3, ae = 3, fg = 3, gh = 3, ij = 3, and jk = 3. For simplicity we can present our list as follows: (1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 3).

Step 2: We break ties arbitrarily and proceed with "highlighting" the corresponding edges cd, bf, kl, ab, cg, fj, bc, ae, gh, ij, and jk. We highlight edge cd first. Then, bf, kl, ab, cg, fj, bc, gh, ij, and jk are highlighted in that order. Edge fg is not highlighted or discarded because it will create a cycle. Since the graph has n = 12vertices, we continued until we have all n - 1 = 11 edges of the graph highlighed. A spanning tree is obtained, and the resulting minimum spanning tree has a total weight of 1 + 1 + 1 + 2 + 2 + 2 + 3 + 3 + 3 + 3 + 3 = 24.

### 2.4.2 Prim's algorithm

Prim's algorithm was originally discovered by the Czech mathematician Vojtech Jarnik in 1930, and it was later rediscovered in 1957 by Robert Prim, an American mathematician.

#### **Description**:

- Begin by choosing any edge with smallest weight (Ties are broken arbitrarily). Then we highlight it.
- 2. Find the edge that is not highlighted with the smallest weight but is connected to one of the endpoints of a highlighted edge. Skip any edge that will produce a circuit.
- 3. Repeat this process until all vertices are adjacent to a highlighted edge; meaning they are all covered.



Figure 2.5: A weighted graph G.

**Example 2.4.2.** Use Prim's algorithm to design a minimum-cost communications network connecting all the twelve computers represented by the graph in Figure 2.5.

We look at Figure 2.5 and apply Prim's Algorithm. Applying Step 1 we can see that the edge representing the distance between Grand Rapids and Kalamazoo has the smallest weight (distance = 56). Therefore, we highlight the edge between Grand Rapids and Kalamazoo and circle the vertex at Grand Rapids and the vertex at Kalamazoo. Applying Step 2 we can see that the edge representing the distance between Grand Rapids and Saginaw (distance = 113) is the smallest weight among the remaining routes connected to either Grand Rapids or Kalamazoo. So, we highlight the edge between Grand Rapids and Saginaw as part of Step 2. We repeat Step 2 until all vertices are incident to a highlighted route. We highlight the edge between Saginaw and Detroit (distance = 98). The smallest route leaving Toledo goes to Detroit (distance = 58). We highlighted it as the last route and note that all cities are reached by a highlighted route. The minimum spanning tree has a total weight of 56 + 113 + 98 + 58 = 325.



Figure 2.6: Some simple graphs P, Q, and R.



Figure 2.7: Some graphs S, T, and U.

## 2.5 Activity: Spanning trees of a graph

#### I. Practice:



Figure 2.8: Some weighted simple graphs V, W, and X.

1. For each of the following graphs P, Q, and R in Figure 2.6, determine the number of its spanning trees subgraphs.

2. For each of the following graphs S, T, and U in Figure 2.7, determine the number of its spanning trees subgraphs.

3. For each of the following graphs V, W, and X in Figure 2.8, determine the minimum spanning tree using:

(i) Prim's algorithm

(ii) Kruskal's algorithm.

#### **II.** Application

1. A company plans to build a communications network connecting its five computer centers as shown in Figure 2.9. Any pair of these centers can be linked with a leased telephone line. Which links should be made to ensure that there is a path between any two computer centers so that the total cost of the network is minimized?

2. The roads represented in the graph in Figure 2.10 by these towns in Nevada are all unpaved. The lengths of the roads between pairs of towns are represented by edge weights. Which minimum road length that guaranties a path of paved roads between each pair of towns?

### 2.6 Answers

#### I. Practice:

San Francisco



Figure 2.9: A Weighted Graph Showing Monthly Lease Costs for Lines in a Computer Network

1. We present the elements of the adjacency and the degree matrices in the order that corresponds to the alphabetical order of the vertices of the graph.(a) Number of spanning trees for graph P.

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}; D = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}; L = \begin{pmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & -1 & 2 & 0 \\ -1 & 0 & -1 & 0 & 2 \end{pmatrix}.$$
  
Now we compute, say,  $M_{11} = det \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & 0 \\ 0 & -1 & 0 & 2 \end{pmatrix} = 12.$ 

(b) Number of spanning trees for graph 
$$Q$$
.  

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}; D = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}; L = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix}.$$
 Now we come



Figure 2.10: A Weighted Graph of some towns in Nevada

pute, say, 
$$M_{11} = det \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 3 \end{pmatrix} = 8.$$

(c) Number of spanning trees for graph 
$$R$$
.  

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}; D = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; L = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}.$$
 Now we com-  
pute, say,  $M_{11} = det \begin{pmatrix} 3 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 1 \end{pmatrix} = 3.$   
2. (a) Number of spanning trees for graph  $S$ .

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}; D = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix}; L = \begin{pmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 3 & -1 & 0 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & -1 & 0 & -1 & 3 \end{pmatrix}.$$
  
Now we compute, say,  $M_{11} = det \begin{pmatrix} 3 & -1 & 0 & -1 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 3 \end{pmatrix} = 24.$ 

(b) Number of spanning trees for graph 
$$T$$
.

$$A = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}; D = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix}; L = \begin{pmatrix} 3 & 0 & -1 & -1 & -1 \\ 0 & 2 & -1 & 0 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}.$$
  
Now we compute, say,  $M_{11} = det \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 4 & -1 & -1 \\ 0 & -1 & 3 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix} = 40.$ 

(c) Number of spanning trees for graph U.

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}; D = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 6 \end{pmatrix}; L = \begin{pmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 4 & 0 & -1 & -1 \\ -1 & 0 & 4 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & -1 & 6 \end{pmatrix}.$$
  
Now we compute, say,  $M_{11} = det \begin{pmatrix} 4 & 0 & -1 & -1 \\ 0 & 4 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 6 \end{pmatrix} = 184.$ 

3. Finding the minimum spanning trees

- (a) Graph V:
- (i) Prim's algorithm

Applying Step 1 we can see that the edge between vertices a and b and the edge between vertices c and d have the smallest weight (ab = 1; cd = 1). We will highlight edge ab and circle vertices a and b. Applying Step 2 we can see that the edge between vertices a and e has the smallest weight (ae = 2) among the remaining unhighlighted edges that have one circled vertex and one uncircled vertex. Therefore, we highlight edge ae and circle the vertex e. Step 3 states to repeat Step 2 until all vertices are circled. Hence, we now highlight edge ed (ed = 2) and circle vertex d. Lastly we highlight edge cd (cd = 1) and circle vertex c. Now every vertex has been circled and connected. Our minimum spanning tree is 1 + 2 + 2 + 1 = 6.

#### (ii) Kruskal's algorithm

Step 1 states to verify that our graph is connected, which we can easily that it is. According to Step 2 we are to create a list: ab = 1, cd = 1; ae = 2, de = 2; bd = 3, be = 3, ce = 3; and ac = 4. For the edges that have the same weight it does not matter which edge you list first. Through Steps 3 and Step 4 we can highlight edges ab, ae, de, and cd. According to Step 3 we highlight edge ab first. Now applying Step 4, edges ae, de, and cd are highlighted in that order. Edges ac, bd, be, and ce are not used because they would create cycles. Since the graph has 5 vertices, Step 4 is continued until we have all n - 1 or 4 edges of the graph highlighted. From Step 5 we can see that we have constructed our minimum spanning tree with a minimal weight: 1 + 2 + 2 + 1 = 6.

- (b) Graph W:
- (i) Prim's algorithm

Applying Step 1 we can see that the edge between vertices e and f has the smallest weight (ef = 1). We will highlight edge ef and circle vertices e and f. Applying Step 2 we can see that the edges between vertices c and f (cf = 3) and e and h (eh = 3) have the smallest weight among the remaining unhighlighted edges that have one circled vertex and one uncircled vertex. Therefore, we highlight edge cf (cf = 3)

and circle vertex c. Step 3 states to repeat Step 2 until all of the vertices are circled. Hence, we see that edge eh (eh = 3) has the smallest weight from the remaining unhighlighted edge with one circled vertex and one uncircled vertex. Therefore, we highlight edge eh and circle vertex h. Continuing the process, edge hi (hi = 2) has the smallest weight from the remaining unhighlighted edges with one circled vertex and one uncircled vertex. We now highlight edge hi and circle vertex i. Edge bc(bc = 2) between vertices b and c and edge gh (gh = 2) between vertices g and h have the smallest weight among the remaining unhighlighted edges with one circled vertex and one uncircled vertex. We will highlight edge bc and circle vertex b. Edge bd(bd = 3) has the smallest weight among the remaining unhighlighted edges with one circled vertex and one uncircled vertex. Thus, we highlight edge bd and circle vertex d. Next, we see that between vertices a and d edge ad (ad = 2) has the smallest weight among the remaining unhighlighted edges with one circled vertex and one uncircled vertex. Hence, we highlight edge ad and circle vertex a. Lastly, we highlight edge qh(gh = 4) and circle vertex g. All vertices of graph W are now circled and connected. Our minimum spanning tree is 1 + 3 + 3 + 2 + 4 + 3 + 2 + 4 = 22.

#### (ii) Kruskal's algorithm

Step 1 states to verify that our graph is connected, which we can easily see that it is. According to Step 2 we are to create a list: ef = 1, ad = 2, hi = 2; bd =3, cf = 3, eh = 3; bc = 4, fh = 4, fi = 4, gh = 4; ab = 5be = 5; bf = 6, dg =6; de = 7, and dh = 8. For the edges that have the same weight, it doesn't matter which edge you list first. Through Step 3 and Step 4 we can highlight edges ef, ad, hi, bd, cf, eh, bc, and gh. According to Step 3 we highlight edge ef first. Now applying Step 4, edges ad, hi, bd, cf, eh, bc, and gh are highlighted in that order. Edges fh, fi, ab, be, bf, dg, de, and dh are not highlighted because they will create cycles. Since graph W has 9 vertices, Step 4 is continued until we have all n - 1 edges or 8 edges of the graph highlighted. From Step 5 we can see that we have constructed our minimum spanning tree with a minimal weight 1+2+2+3+3+3+4+4=22.

- (c) Graph X:
- (i) Prim's algorithm

Applying Step 1 we can see that the edge between vertices a and b, (ab = 1), the edge between vertices a and e (ae = 1), the edge between vertices c and d, (cd = 1), and the edge between vertices d and h (dh = 1) have the same smallest weight. We will highlight edge ab and circle vertices a and b. Applying Step 2 we can see that the edge between vertices a and e (ae = 1) is the smallest weight among the remaining unhighlighted edges that have one circled vertex and one uncircled vertex. Therefore, we now highlight edge ae and circle vertex e. Step 3 states to repeat Step 2 until all vertices are circled. Hence, we now highlight edge bc, (bc = 2) and circle vertex c from the remaining unhighlighted edges with one circled vertex and one uncircled vertex. Next, edge cd (cd = 1) is highlighted and vertex d is circled. Continuing, edge dh (dh = 1) is highlighted and vertex h is now circled. Edge gh, (gh = 2), is now highlighted and vertex g is circled. Next, edge ef, (ef = 2), is highlighted and vertex f is circled. Edge  $e_i$ ,  $(e_i = 2)$ , is now highlighted and vertex *i* is circled. Edge ij, (ij = 3), is now highlighted and vertex *j* is circled. Continuing the process, edge im, (im = 3) is highlighted and vertex m is circled. Next, edge mn, (mn = 2), is highlighted and vertex n is circled. Edge no, (no = 2), is now highlighted and vertex o is circled. Edge op, (op = 3) is highlighted next and vertex p is circled. Edge lp, (lp = 2), is highlighted and vertex l is now circled. Lastly, edge kl, (kl = 3) is highlighted and vertex k is circled. Now all of the vertices of graph X have been circled and connected yielding a minimum spanning tree of 1 + 1 + 2 + 1 + 1 + 2 + 2 + 2 + 3 + 3 + 2 + 2 + 3 + 2 + 3 = 30.

#### (ii) Kruskal's algorithm

Step 1 states to verify that graph X is connected, which we can easily see that it is. According to Step 2 we are to create a list: ab = 1, ae = 1, cd = 1, and dh = 1; ad = 2, am = 2, bc = 2, dp = 2, mp = 2, gh = 2, ef = 2, ei = 2, mn = 2, no =2, lp = 2; bf = 3, cg = 3, fg = 3, fj = 3, ij = 3, im = 3, hl = 3, kl = 3, ko = 3, op = 3;and gk = 4, jk = 4, and jn = 4. For the edges that have the same weight, it doesn't matter which edge you list first. Through Step 3 and Step 4 we can highlight edges ab, ae, bc, cd, dh, gh, ef, ei, ij, im, mn, no, op, lp, and kl. According to Step 3 we highlight edge ab first. Now applying Step 4, edges ab, ae, bc, cd, dh, gh, ef, ei, ij, im, mn, no, op, lp, and kl are highlighted in that order. Edges ad, am, dp, mp, bf, cg, fg, fj, hl, koand jn are not highlighted because they will create cycles. Since graph X has 16 vertices, Step 4 is continued until we have all n-1 or 15 edges of the graph highlighted. From Step 5 we can see that we have constructed our minimum spanning tree with a minimal weight 1+1+2+1+1+2+2+2+3+3+2+2+3+2+3=30. Figure 2.11 shows an example of a minimum spanning tree for the graph X.



Figure 2.11: A minimum spanning tree of Graph X from Figure 2.8

#### **II.** Applications:

1. This problem is applicable to finding a minimum spanning tree. As such we will use Prim's algorithm for Figure 19 to find which links should be made to ensure that there is a path between any two computer centers so that the total cost of the network is minimized. Applying Step1 we can see that the link between Chicago and Atlanta has the lowest cost of amount of \$350. This edge (link) is highlighted and vertices Chicago and Atlanta are circled. Applying Step 2, the edge between Atlanta and New York has the lowest cost, \$400, of the remaining unhighlighted edges with one circled vertex and one uncircled vertex and is therefore, highlighted and the New York vertex is circled. Continuing the process, the Chicago to San Francisco link, \$600, is highlighted and vertex San Francisco is circled. Finally, we highlight the San Francisco to Denver link, \$450, and circle vertex Denver. All vertices are now circled and connected producing a minimum spanning tree with a minimal cost of \$350 + \$400 + \$600 + \$450 = \$1800.

2. We use Prim's algorithm to find a path of paved roads between each pair of towns with a minimum road length. The solution is as follows: Deep Springs – Oasis = 10; Oasis – Dyer = 21; Oasis - Silver Pea = 23; Silver Pea – Goldfield = 20; Goldfield – Lida = 20; Lida – Gold Point = 12; Goldfield – Tonopah = 35; Tonopah

- Manhattan = 25; Beatty - Goldpoint = 45; and Tonopah - Warm Springs = 55. This pathway of roads is our minimum spanning tree with a total road length of 10 + 21 + 23 + 20 + 20 + 12 + 35 + 25 + 45 + 55 = 266.

### Chapter 3 Eulerian and Hamiltonian Graphs

This chapter is designed to prepare the student for NCSBE competency goal 1– objective **1.0.1.c**.

The original Konisberg problem depicted in Chapter 1.3 (see Figure 1.2) requires that the city is "Eulerian", if modeled as a (multi)graph. We define this concept here and describe Euler's answer.

## **3.1** Eulerian Graph and Cycle

An **Euler cycle** or circuit in a graph G is a simple cycle containing every edge (exactly once) of G. An **Euler path** in G is a simple path containing every edge (exactly once) of G.

The following two facts are useful in determining whether or not a graph has an Euler cycle or path.

**Fact1:** A connected graph has an Eulerian cycle or circuit if and only if every vertex is of even degree.

**Fact2:** A connected graph has an Eulerian path if and only if it has exactly two vertices of odd degrees.

So, Euler simply proved that the inhabitants of Konisberg cannot start from a bank or an island and walk accross all the bridges exactly once and return to their starting bank because at least one of the banks has an odd degree; clearly, as shown in Figure 1.2, all the banks have odd degrees. A proposed solution for this city is to build a minimum of two new bridges. One bridge between Right Bank and Left Bank, bringing their degree to 4, and one additional bridge between Island and East Side, bringing their degrees to 6 and 4, respectively. Thus, satisfying the stated condition in Fact 3.1.



Figure 3.1: Some simple graphs named  $G_1$ ,  $G_2$ ,  $G_3$  (from left to right).

Example 3.1.1. Consider the graphs in Figure 3.1.

- (i) G<sub>1</sub> contains exactly two vertices of odd degree (b and d). Hence it has an Euler path, e.g., listing only the order in which the vertices are being visited, we have d, a, b, c, d, b.
- (ii)  $G_2$  has exactly two vertices of odd degree (b and d). For instance, listing only the order in which the vertices are being visited, we have b, a, g, f, e, d, c, g, b, c, f, d.

(iii)  $G_3$  has six vertices of odd degree. Hence, it does not have an Euler path.

Note that because none of the graphs has only vertices of even degrees, none has an Euler cycle. An example of a graph that has an Euler cycle, see Figure 1.3. Clearly all of its vertices are of even degrees.

## **3.2** Hamiltonian Graph and Cycle

A Hamiltonian path is a path that visits each vertex of a graph exactly only once. A Hamiltonian cycle is a Hamiltonian path that starts and ends at the same vertex.

A very important graph theory application is called the **Traveling Salesperson Problem (TSP)**. Given a weighted graph, a TSP asks for the cycle of minimum total weight which visits each vertex exactly once and returns to its starting point. So in fact, TSP is equivalent to finding a Hamilton cycle with minimum total weight in a graph.

**Note:** Euler paths and cycles contained every edge only once while Hamiltonian paths and cycles that contain every vertex exactly once.

**Example 3.2.1.** Consider the graphs in Figure 3.2.



Figure 3.2: Some simple graphs named  $G_4$ ,  $G_5$ ,  $G_6$  (from left to right).

- (i) G<sub>4</sub> does not have a Hamilton cycle (Why?), but does have a Hamilton path, say,
   a, b, e, d, c.
- (ii)  $G_5$  has a Hamilton cycle, say, a, b, c, d, e, a.
- (iii)  $G_6$  has a Hamilton cycle, say a, b, e, d, c, a.

**Note:** Unlike for an Euler circuit, no simple necessary and sufficient conditions are known for the existence of a Hamilton circuit. We state this condition here:

**Dirac's Theorem (necessary condition)**: If G is a simple graph with  $n \ge 3$  vertices such that the degree of every vertex in G is at least  $\lceil n/2 \rceil$ , then G has a Hamilton circuit.

Observe that  $G_6$  has a Hamilton cycle even though it does not satisfy the necessary condition of Dirac's Theorem; i.e., not all of its vertices are of degree  $3 \ge 3 = \lceil 5/2 \rceil$ . Also, it is easy to notice that every complete graph on n vertices admits a Hamilton cycle.

# 3.3 Activity

#### I. Practice

For each of the graphs in Figure 3.3, find the following:

(a) determine whether the given graph has an Euler circuit. Construct such a circuit when one exists. If no Euler circuit exists, determine whether the graph has an Euler path and construct such a path if one exists.

(b) determine whether the given graph has a Hamilton circuit. If it does, find such a circuit. If it does not, give an argument to show why no such circuit exists.

#### **II.** Application



Figure 3.3: Some simple graphs named A, B, C and D (from top-left to bottom-right).



Figure 3.4: A floor plan

1. The floor plan shown in Figure 3.4 is for a house that is open for public viewing. Is it possible to find a trail that starts in room A, ends in room B, and passes through every interior doorway of the house exactly once? If so, find such a trail.

2. Consider the graph in Figure 2.5. Suppose that a salesman salesperson wants to visit all the five cities, Detroit, Toledo, Saginaw, Grand Rapids, and Kalamazoo exactly once, starting and ending in city A. In which order should he visit these cities to travel the minimum total distance?

3. Find a route with the least total airfare that visits each of the cities of the graph in Figure 3.5, where the weight on an edge is the least price available for a flight between the two cities. What is the total flights cost?



Figure 3.5: Flights cost between each pair of cities

### 3.4 Answers

#### I. Practice:

(a)

**Graph A:** To be an Euler circuit the degree of every vertex must be even. Vertices c and f have odd degrees; deg(c) = 3, deg(f) = 3. Therefore, Graph A has no Euler circuit. Since Graph A has at most 2 vertices of odd degree, it does have an Euler path. One such path is ca - ab - bc - cf - fe - ed - df.

**Graph B:** Since not every vertex has an even degree, deg(a) = 3, deg(b) = 3, deg(c) = 3, deg(f) = 1 it does not have an Euler circuit. Also, because Graph *B* has more than 2 vertices of odd degree, it has no Euler path.

**Graph C:** Not every vertex has an even degree, deg(b) = 3, deg(d) = 3, deg(f) = 3, deg(f) = 3, deg(j) = 3. So, Graph C does not have a Euler circuit. Further, Graph C has more than 2 vertices of odd degree, hence, it has no Euler path.

**Graph D:** Not every vertex has an even degree, deg(b) = 3, deg(d) = 3, deg(f) = 3, deg(h) = 3, so, it does not have an Euler circuit. Further, Graph D has more than 2 vertices of odd degree therefore, it does not have an Euler path.

(b)

**Graph A:** Graph A does not have a Hamiltonian circuit because if we start with any vertex on either side of vertices c or f we cannot return to our starting point without repeating vertices. Graph A does have a Hamiltonian path: ab - bc - cf - fe - ed. **Graph B:** There is no possibility of a Hamiltonian circuit in Graph *B* because once you go to vertex *f* there is no way to leave without returning to vertex *e*. Graph *B* does have a Hamiltonian path: fe - eb - bc - ca - ad.

**Graph C:** According to Dirac's Theorem, if G is a simple graph with n vertices with  $n \ge 3$  such that the degree of every vertex in G is at least n/2, then G has a Hamiltonian circuit. Graph C has 17 vertices. Every vertex in Graph C has a degree smaller than 17/2. Therefore, it does not have a Hamiltonian circuit.

**Graph D:** Starting at vertex e, then move to each vertex in the order to vertex b to vertex a to vertex d to vertex g to vertex h to vertex i to vertex f to vertex c then back to vertex e, we see that Graph D does have a Hamiltonian circuit.

#### II. Application

1. Let the floor plan of the house be represented by the graph in Figure 3.6.



Figure 3.6: Floor plan graph

Each vertex of this graph in Figure 3.6 has even degree except for A and B, each of which has degree 1. Hence by Fact 3.1, there is an Euler path from A to B. One such trail is AGHFEIHEKJDCB.

2. To solve this problem we can assume the salesperson starts in Detroit (because this must be part of the circuit) and examine all possible ways for him to visit the other four cities and then return to Detroit (starting elsewhere will produce the same circuits). There are a total of 4!/2 = 12 different routes/circuits as shown in Table 3.1: From this table, the minimum total distance of 458 miles (shown in bold) is traveled using the circuit Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit (or its reverse).

Route	Total Distance
Detroit–Toledo–Grand Rapids–Saginaw–Kalamazoo–Detroit	610
Detroit–Toledo–Grand Rapids–Kalamazoo–Saginaw–Detroit	516
Detroit–Toledo–Kalamazoo–Saginaw–Grand Rapids–Detroit	588
Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit	458
Detroit–Toledo–Saginaw–Kalamazoo–Grand Rapids–Detroit	540
Detroit–Toledo–Saginaw–Grand Rapids–Kalamazoo–Detroit	504
Detroit–Saginaw–Toledo–Grand Rapids–Kalamazoo–Detroit	598
Detroit–Saginaw–Toledo–Kalamazoo–Grand Rapids–Detroit	576
Detroit–Saginaw–Kalamazoo–Toledo–Grand Rapids–Detroit	682
Detroit–Saginaw–Grand Rapids–Toledo–Kalamazoo–Detroit	646
Detroit–Grand Rapids–Saginaw–Toledo–Kalamazoo–Detroit	670
Detroit–Grand Rapids–Toledo–Saginaw–Kalamazoo–Detroit	728

Table 3.1: List of different routes for the salesperson starting from Detroit

3. Consider (by a brute force) all possible ways to visit these cities. We obtain that, the itinerary  $SanFrancisco \rightarrow Denver \rightarrow Detroit \rightarrow NewYork \rightarrow LosAngeles \rightarrow$ SanFrancisco (or its reverse) gives the least total flights cost of 179 + 229 + 189 + 379 + 69 = 1055.

#### Remark:

The general traveling salesman problem involves finding a Hamiltonian circuit to minimize the total distance traveled for an arbitrary graph with n vertices in which each edge is marked with a distance.

One way to solve the general problem is to by a brute force like the one we just used for the previous answers 2 and 3; we write down all Hamiltonian circuits starting and ending at a particular vertex, compute the total distance for each, and pick one for which this total is minimal. However, even for relatively small values of n, this method is impractical. For a complete graph on n vertices, we will need to examine exactly (n-1)!/2 such circuits. So, for a complete graph on 30 vertices, for instance, there would be  $29!/2 \simeq 4.42 \times 10^{30}$  Hamiltonian circuits starting and ending at a particular vertex to check. As of today, there is no known algorithm for solving efficiently the general traveling salesman problem.

### Chapter 4 Vertex Coloring

This Chapter is designed to prepare the student for NCSBE competency goal 1– objectives **1.0.1.c**.



Figure 4.1: A 3-colorable planar graph.

A coloring of a graph G is an assignment of a color to each vertex of G so that adjacent vertices receive different colors. Such coloring is said to be **proper**. The **chromatic number** of G is the minimum number of colors needed for a *proper* coloring of G. We often denote the chromatic number of a graph G by  $\chi(G)$ . (To be read "chi of G".) It is easy to see that, for a cycle on n vertices,  $C_n$ ,

$$\chi(C_n) = \begin{cases} 2 & \text{if } n \text{ is even} \\ \\ 3 & \text{if } n \text{ is odd.} \end{cases}$$

Also,  $\chi(K_n) = n$  for any complete graph  $K_n$ .  $\chi(T_n) = 2$  for any tree  $T_n$ .

The notion of graph (vertex) coloring stems from a question asked in 1850's by a South African mathematician, Francis Guthrie, while trying to color the map of counties of England. It is well-known as the **four-color problem** which we restate as follows:

Can we color the maps of any region on the planet using 4 colors such that any two countries with a common border are assigned different colors?

The answer to this question is generally accepted to be "yes", although most solutions depend on computer algorithms. Graphs which admit a proper coloring using k-colors, are said to be k-colorable. Figure 4.1 shows a graph which is 3-colorable, using colors red, blue and green.

**Example 4.0.1.** Find  $\chi(G)$  and  $\chi(H)$  for the graphs shown in Figure 4.2. Assume G is on the left side while H is on the right side.



Figure 4.2: Two simple graphs G and H.

The chromatic number of G is at least three, because the vertices a, b, and c must be assigned different colors. To see if G can be colored with three colors, assign red to a, blue to b, and green to c. Then, d can (and must) be colored red because it is adjacent to b and c. Furthermore, e can (and must) be colored green because it is adjacent only to vertices colored red and blue, and f can (and must) be colored blue because it is adjacent only to vertices colored red and green. Finally, g can (and must) be colored red because it is adjacent only to vertices colored red and green. Finally, g can (and must) be colored red because it is adjacent only to vertices colored blue and green. This produces a coloring of G using exactly three colors, i.e.,  $\chi(G) = 3$ . Figure 4.3 displays such a coloring.

The graph H is made up of the graph G with an edge connecting a and g. Any attempt to color H using three colors must follow the same reasoning as that used to color G, except at the last stage, when all vertices other than g have been colored. Then, because g is adjacent (in H) to vertices colored red, blue, and green, a fourth color, say brown, needs to be used. Hence, H has a chromatic number equal to 4, i.e.,  $\chi(H) = 4$ . We also show a coloring of H in Figure 4.3.

**Note:** Due to the limit on our use of natural colors, it is customary to use integers or natural numbers to denote colors assigned to vertices instead.



Figure 4.3: Two properly colored simple graphs G and H.

# 4.1 Activity

#### I. Practice

1. Find the chromatic number of each graph in Figure 3.3.

2. What is the chromatic number of each of the following graphs  $W_n$ ,  $K_{m,n}$ , for any m, n?

#### **II.** Application

Table 4.1: Distance between pairs of 6 six radio stations

	1	2	3	4	5	6
1		85	175	200	50	100
2	85		125	175	100	160
3	175	125		100	200	250
4	200	175	100		210	220
5	50	100	200	210		100
6	100	160	250	220	100	

1. The chair of the math department meets with six committees, once a month. How many different meeting times must be used to ensure that no member is scheduled to attend two meetings at the same time if the committees are:

 $C_1 = \{Allagan, Sengupta, Talukder\}, C_2 = \{Sengupta, Ogaja, Kulkarni\},$ 

 $C_{3} = \{Allagan, Kulkarni, Talukder\}, C_{4} = \{Ogaja, Kulkarni, Talukder\}, C_{4} =$ 

 $C_5 = \{Allagan, Sengupta\}, \text{ and } C_6 = \{Sengupta, Kulkarni, Talukder\}?$ 

2. Schedule the final exams for Math 114, Math 115, Math 118, Math 165, CS 111, CS 114, CS 115, and CS 215, using the fewest number of different time slots, if there are no students taking both Math 114 and CS 215, both Math 115 and CS 215,

both Math 165 and CS 111, both Math 165 and CS 114, both Math 114 and Math 115, both Math 114 and Math 118, and both Math 118 and Math 165, but there are students in every other pair of courses.

3. How many different channels are needed for six stations located at the distances shown in Table 4.1, if two stations cannot use the same channel when they are within 150 miles of each other?

## 4.2 Answers

#### I. Practice:

1.

**Graph A:**  $\chi(A) = 3$ ; The chromatic number of Graph A is at least three. Vertices a, b, and c must be assigned different colors. Hence, a sample coloring is: a is colored red, b is colored blue, and c is colored green. Now, vertices d, e, and f can be colored with the same three colors. However, vertex f must have a different color from vertex c because they are adjacent vertices. Therefore, since c is colored green then f could be colored red or blue. Finally, vertices d and e will be colored differently with the two remaining colors. Here is a 3-coloring: a = red, d = green, b = blue, e = blue, c = green, f = red.

**Graph B:**  $\chi(B) = 3$ ; The chromatic number of Graph *B* is at least three. Vertices *a*, *b*, and *c* must each be assigned different colors. Hence, a sample coloring is: *a* is colored red, *b* is colored blue, and *c* is colored green. Now, *e* has to be colored differently from vertices *b* and *c* because they are adjacent vertices. Hence, *e* would be colored red just like vertex *a*. Finally, vertices *d* and *f* must be colored differently from vertices *a* and *e*. Vertices *d* and *f* can be colored the same colors because they are nonadjacent or they can be colored differently. Hence, vertices *d* and *f* can be colored blue or green. Here is a 3-coloring of the graph *G*. *a* = red, *d* = blue or green, *b* = blue, *e* = red, *c* = green, *f* = blue or green.

**Graph C:**  $\chi(C) = 2$ ; The chromatic number of Graph C is at least two. The color of each vertex is alternated between two colors. So, starting with vertex a, we color it

red then b is colored blue; c is colored red then h is colored blue; g is colored red then f is colored blue; e is colored red then d is colored blue. Since vertex o is adjacent to vertex d, vertex o must be colored red then p is colored blue and q is colored red. We know this is correct because q is adjacent to h which we colored blue. Now adjacent to vertex o, vertices i and n can both be colored blue. Vertices j and m are adjacent to p which is colored blue, so, they are red. Vertices k and l are adjacent to q which is colored red. Therefore, they are colored blue. Every adjacent vertex has a different color alternating between red and blue. Here is a 2-coloring of the graph C: a = red, b = blue, c = red, h = blue, g = red, f = blue, e = red, d = blue, o = red, p = blue, q = red, m = red, n = blue, j = red, l = blue, k = blue.

**Graph D:**  $\chi(D) = 2$ ; The chromatic number of Graph D is at least two. Vertices a, c, e, g, and i can be colored the same. The remaining vertices b, d, f, and h are colored the same color together but, different from vertices a, c, e, g, and i. Therefore, a sample coloring is: a, c, e, g, and i are colored red while b, d, f, and h are colored blue.  $a = red, \ b = blue, \ c = redd = blue, \ e = red, \ f = blue, \ g = red, \ h = blue, \ i = red.$ 2.

(i.) For any Wheel on n vertices,

$$\chi(W_n) = \begin{cases} 3 & \text{if } n \text{ is even} \\ 4 & \text{if } n \text{ is odd.} \end{cases}$$

(ii.) For any complete bipartite graph,  $\chi(K_{m,n}) = 2$  for all m, n.

#### **II.** Applications:

1. To determine the minimal number of different meeting times needed to ensure that no member is scheduled to attend 2 meetings we draw the graph in Figure 2.11, where vertices represent committees and two vertices are adjacent if the corresponding committees share a member. For instance, Committee 1,  $C_1$ , has members that are also on every other committees,  $C_2$  through  $C_6$ . So, there is an edge connecting  $C_1$ to  $C_2$ ,  $C_3$ ,  $C_4$ ,  $C_5$ , and  $C_6$ . The same is true for all other vertices, except  $C_4$  and  $C_5$ , which are adjacent to other vertices except to each other. The vertices can be colored as follows:  $C_1 = \text{red}$ ,  $C_2 = \text{blue}$ ,  $C_3 = \text{yellow}$ ,  $C_4 = \text{green}$ ,  $C_5 = \text{green}$ , and  $C_6$  = purple. A minimum of 5 different meeting times is needed to ensure that every person can attend all of their required meetings with no conflict.



Figure 4.4: Committee Graph showing committees that share a member in common.

2. To schedule 8 exams so there is no conflict for students, a graph is drawn as shown in Figure 4.5 where each vertex represents an exam and an edge connects two vertices if no student takes those two courses together. For instance, an edge connects Math 114 with Math115, Math 118 and CS 215 because there were no students taking both Math 114 and Math 115, both Math 114 and Math 118, and both Math 114 and CS 215. The absence of an edge between two courses is an indication that at least a student is enrolled in these two courses at the same time. So, we color the vertices so that non-adjacent vertices are given different colors; in which case adjacent vertices may be given same color. This graph (conflict-free) is known as a *complementary* graph to a graph which would show the cases where adjacent vertices indicate conflict. Note that both graphs, together, form  $K_8$ , a complete graph on 8 vertices.



Figure 4.5: Courses Graph with no scheduling conflict.

Here is a coloring: Math 114, Math 115, and CS 215 – red; Math 165 and CS 114

- blue; CS 111 - green; CS 115 - yellow; and Math 118 - purple.

This previous coloring is equivalent to the following a possible scheduling:

- $1^{st}$  Period (red) M 114, M 115, CS 215
- $2^{nd}$  Period (blue) M 165 and CS 114
- $3^{rd}$  Period (green) CS 111
- $4^{th}$  Period (yellow) CS 115
- $5^{th}$  Period (purple) Math 118.

3. In order to determine how many channels are needed for the 6 stations, we draw the graph shown in Figure 4.6 to model the problem. Each station is represented by a vertex. Two vertices are adjacent if the stations are within 150 miles of each other. Station 1 is adjacent to Stations 2, 5, and 6. Station 2 is adjacent to Stations 1, 3, and 5. Station 3 is adjacent to Stations 2 and 4. Station 4 is adjacent to Station 3. Station 5 is adjacent to Stations 1, 2, and 6. Station 6 is adjacent to Stations 1 and 5. Here is a 3-coloring as shown in Figure 4.6. Station 1 = red, Station 2 = blue, Station 3 = red, Station 4 = blue, Station 5 = green, and Station 6 = blue.



Figure 4.6: Radio stations located within 150 miles or less.

### Chapter 5 Conclusion and Future Research

In this thesis, we introduced students to three fundamental research topics commonly discussed in Graph Theory, with the goal of strengthening their math skills in the required competency goal 1 in North Carolina. Chapters presented include Spanning trees which covers competency objectives 1.0.1.a,b, Euler, Hamilton Graphs and Vertex Coloring cover competency objective 1.0.1.c. Most of the activities are relatively simple and easy to follow. We recommend students to work on Practice problems first, and then move onto the Applications. Students can begin any other chapter, after completing Chapter 1. A pre-requisite knowledge of how to compute, up to, a  $4 \times 4$  determinant is required for Chapter 4 (Spanning trees). Students often find it difficult to compute the determinant of graphs of order greater than 5, that is why most of our problems are limited to graphs with order 6 or less. However, with some counting skills, we think that they can arrive at some of the results of finding the number of spanning trees of graphs without resulting to Kirchhoff's theorem [1] which relies on the computation of determinants. We recommend that future work include other relevant topics in graph theory such as Chromatic Polynomials which would target other competency goals and learning objectives. We close the thesis, with the proposed conjecture, which can lead to further results on counting spanning trees.

A **cactus** is a simple connected graph in which every pair of cycles share at most one vertex. We conjecture the following,

**Conjecture 5.0.1.** Suppose G is a cactus with k cycles of orders  $m_1, m_2, \ldots, m_k$ . Then the number of its spanning trees is  $\prod_{i=1}^k m_i$ .

### Bibliography

- [1] Buekenhout, F. and Parker, M. The Number of Nets of the Regular Convex Polytopes in Dimension  $\leq 4$ , Disc. Math., **186**, 69-94, 1998.
- [2] GAIMME-Guidelines for Assessment & Instruction in Mathematical Modeling Education. Retrieved at http://www.siam.org/Portals/0/ Publications/Reports/gaimme-full\_color\_for\_online\_viewing.pdf? ver=2018-03-19-115454-057.
- [3] K-12 Standards, Curriculum and Instruction. Retrieved at http://www.dpi. state.nc.us/curriculum/
- [4] 2018-2019 NC Final Exam of Discrete Mathematics North Carolina Assessment Specifications. Retrieved at http://www.ncpublicschools.org/docs/accountability/testing/ common-exams/specifications/2018-19/dscrtmthspecs18.pdf
- [5] K. H. Rosen, *Discrete Mathematics and its Applications*, McGraw Hill, New York, NY, 2019.
- [6] D. B. Smithers, Graph Theory for the Secondary School Classroom, MS thesis, East Tennessee State University, 2005. Retrieved at https://dc.etsu.edu/cgi/ viewcontent.cgi?article=2172&context=etd.

## Appendix

- 1. F-LE.– Functions:
  - Linear, Quadratic, and Exponential Models
  - Construct and compare linear and exponential models and solve problems: NC.M1 and NC.M2 (F-LE. 1 – 5)
- 2. A-CED.– Algebra: Equations
  - Create equations that describe numbers or relationships
  - Create inequalities in one variable that represent absolute value, polynomial, exponential, and rational relationships and use them to solve problems algebraically and graphically: NC. M1 and NC. M2 (A-CED. 1 3)
- 3. S-ID.– Statistics and Probability
  - Interpreting Categorical and Quantitative Data
  - Summarize and interpret linear models: NC. M1 (S-ID. 7 9)
- 4. GAIMME Guidelines For Assessment & Instruction In Mathematical Modeling Education
- 5. NC.M1.– North Carolina Math I
- 6. NC.M2.– North Carolina Math II
- 7. NCSCOS. North Carolina Standard Course Of Study