EDGE COLORING OF A GRAPH

Tina BEŞERİ

July, 2004

Edge Coloring of a Graph

By

Tina BEŞERİ

A Dissertation Submitted to the
Graduate School in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

Department: Mathematics
Major: Mathematics

İzmir Institute of Technology
İzmir, Turkey

June, 2004

We approve the thesis of **Tina BEŞERİ**

Date of Signature

———————————————————   22.07.2004

Assist.Prof.Dr. Ünal UFUKTEPE

Supervisor

Department of Mathematics

———————————————————   22.07.2004

Assoc.Prof.Dr. Alpay KIRLANGIÇ

Department of Mathematics

Ege University

———————————————————   22.07.2004

Assist.Prof.Dr. Murat ATMACA

Department of Mathematics

———————————————————   22.07.2004

Assist.Prof.Dr. Gamze TANOĞLU

Head of Department of Mathematics

ACKNOWLEDGEMENTS

ABSTRACT

The edge coloring problem is one of the fundamental problem on graphs which often appears in various scheduling problems like the file transfer problem on computer networks. In this thesis, we survey old and new results on the classical edge coloring as well as the generalized edge coloring problems. In addition, we developed some algorithms and modules by using Combinatorica package to color the edges of graphs with web*Mathematica* which is the new web-based technology.

ÖZ

Bilgisayar ağlarındaki dosya transfer problemleri gibi birçok farklı zamanlama probleminde de sık sık ortaya çıkan çizgelerdeki en temel problemlerden biri kenar boyama problemidir. Bu tezde, genelleşmiş kenar boyama problemlerinde bugüne kadar elde edilen eski ve yeni sonuçları araştırdık. Buna ek olarak web tabanlı yeni bir teknoloji olan web*Mathematica* ile çizgelerin kenarlarını boyamak için Combinatorica yazılımını kullanarak bazı algoritmalar ve modüller geliştirdik.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

A graph is an abstract structure which consists of vertices and edges; each edge joins two vertices called ends of the edge. It can be used to represent various combinatorial or topological structures that can be modelled as objects and connections between those objects. A graph structure is very suitable for representing relationships between objects in the abstract, and a large number of combinatorial problems can be modelled as problems on the graph structure.

Consider the following simple example. Suppose that there are several boys and girls, and each girl favors some of the boys. We want to make a date-matching in which every girl is assigned to one of her favorite boys. Consider a graph in which boys and girls correspond to vertices and a vertex corresponding to a girl is joined by edges to vertices corresponding to boys whom the girl favors. (See Figure 1.1.) We can then make the desired date-matching by finding a set M of edges called a matching satisfying that each vertex corresponding to a girl is an end of an edge in M, and no two edges in M share a common end. In Figure 1.1, the edges in a matching M are indicated by solid lines, and the remaining edges are indicated by dotted lines. This problem is called the date-matching problem, and it can be solved by using a concept of a matching of graphs, as above. There are a numerous number of situations to which the graph theory can be validly applied like this example.

Partitioning a set of objects into some classes according to certain rules is a fundamental process in mathematics, and it appears in many actual situations. A conceptually simple set of rules tells us for each pair of objects whether or not they are allowed to belong to the same class. Of course, graph theory has a powerful tool to deal with such a situation, a concept of graph coloring. In the sense of not only applicative interest but theoretical interest, the graph coloring theory is one of the most attractive field of graph theory indeed.
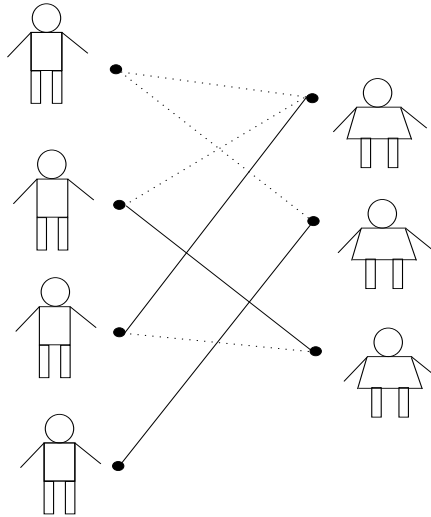
Figure 1.1: A date-matching problem

The Theory of Graph Coloring has existed for more than 150 years. From its modest beginning of determining whether a geographic map can be colored with four colors, the theory has become central in Discrete Mathematics with many contemporary generalizations and applications. Historically, graph coloring involved finding the minimum number of colors to be assigned to the vertices / edges / regions so that adjacent vertices / edges / regions must have different colors.

We start in Chapter 2 by giving the basic and standard terminologies and notations on graph theory which will be used in this thesis. In Chapter 3, we introduce the edge coloring problem. In Section 3.1, we present an historical account of the development of the theory of edge-coloring from Tait's (1880) paper to the classical work of Vizing and recent ones by others. In the next Section, we define edge coloring and chromatic index of a graph, then in Section 3.3, we introduce the concept of Line Graph. Chromatic Index for Common Graph Families is given in Section 3.4. In Section 3.5, we give the definitions of Chromatic Incidence. We present proofs of the fundamental theorems of König and Vizing in Section 3.6 and 3.7. Chromatic Index for Multigraphs is given in Section 3.8. Then, in Section 3.9, we introduce the Edge Coloring Problem for Planar Graphs. Edge Game Coloring of Graphs are defined in Section 3.10. Finally, in Chapter 4, we describe some web-based interactive examples on edge coloring with web*Mathematica*.

Chapter 2

PRELIMINARIES

In this chapter, we provide the necessary background and motivation for this study on the edge-coloring of graphs. Further explanation of these terms can be found in any of the standard texts in graph theory [2, 6, 9]. We start in Section 2.1 by giving some definitions of standard graph-theoretical terms used throughout the remainder of the thesis. We next define common graph families in graph theory in Section 2.2. Then, in Section 2.3 we introduce the Eulerian and Hamiltonian Graphs. Finally, in Section 2.4 we define the NP-Complete Problem. Definitions which are not included in this section will be introduced as they are needed.

2.1   Concepts of Graphs

A **graph** $G = (V, E)$ consists of two sets: a non-empty finite set V and a finite set E. The elements of V are called **vertices** (or points or nodes) and the elements of E are called **edges** (or lines). Each edge is identified with a pair of vertices. The set $V(G)$ is called the **vertex set** of G, and the set $E(G)$ is called the **edge set** of $E(G)$. If $e = \{u, v\} \in E(G)$ then we say that e **joins** u and v. The vertices u and v are called the **ends** of the edge uv. The **order** of a graph, denoted by $n(G)$, is the number of vertices, and the **size** of a graph, denoted by $m(G)$, is the number of edges. Graphs are finite or infinite according to their order; however the graphs we consider are all finite. If a graph allows more than one edge (but yet a finite number) between the same pair of vertices in a graph, the resulting structure is a **multi-graph**. Such edges are called **parallel** or **multiple edges**. An edge that joins a single endpoint to itself is known as a **loop**. Graphs that allow parallel edges and loops are called **pseudographs**. A **simple graph** is a graph with no parallel edges and loops.

A graph H is called a **subgraph** of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A subgraph H of G is said to be an **induced subgraph** of G if each edge of G having its endpoints in $V(H)$ is also an edge of H. A subgraph H of G is a

spanning subgraph of G, if $V(H) = V(G)$. A **directed graph** (or digraph) is a graph each of whose edges is directed.

We say that two vertices u and v of a graph G are **adjacent** if there is an edge uv joining them, and the vertices u and v are then **incident** with such an edge. Similarly, two distinct edges e and f are adjacent if they have a vertex in common. An **independent set of edges** in a graph G is a set of edges, each two of which are not adjacent. The **edge independence number** $\beta_1(G)$ of G is the maximum cardinality among the independent sets of edges of G.

Let G be a graph and $v \in V$. The number of edges incident at v is called the **degree** of the vertex v in G and denoted by $deg(v)$. A loop at v is to be counted twice in computing the degree of v. A vertex of degree 0 is called an **isolated vertex**. A graph G is **regular** if all the vertices of G are of equal degree. If every vertex of G has degree r, then G is called **r-regular**. $\delta(G) = min\{deg(v)|v \in V\}$ denotes the **minimum degree** of G. Similarly, $\Delta(G) = max\{deg(v)|v \in V\}$ denotes the **maximum degree** of G. The **degree sequence** of a graph is the sequence formed by arranging the vertex degrees in increasing order. Although each graph has a unique degree sequence, two structurally different graphs can have identical degree sequences.

Proposition 1. *If G is a graph with n vertices, then for any vertex v,*

$$0 \leq deg(v) \leq n - 1.$$

Proposition 2. *A simple graph G must have at least one pair of vertices whose degrees are equal.*

Let G be a simple graph. Then, the **complement** of a graph G, denoted by $\overline{G}$, is the graph with the same vertex set as G, and where distinct vertices x and y are adjacent in $\overline{G}$ if and only if they are not adjacent in G.

2.2 Common Graph Families

A simple graph G is said to be **complete** if every pair of distinct vertices of G are adjacent in G. It is denoted by K_n. A simple graph with n vertices can have at most $\frac{n(n-1)}{2}$ edges. K_n has the maximum number of edges among all simple graphs with n vertices. Thus, for a simple graph G with n vertices we have $0 \leq m(G) \leq \frac{n(n-1)}{2}$. Figure 2.1 represents K_1, K_2, K_3, K_4 *and* K_5.
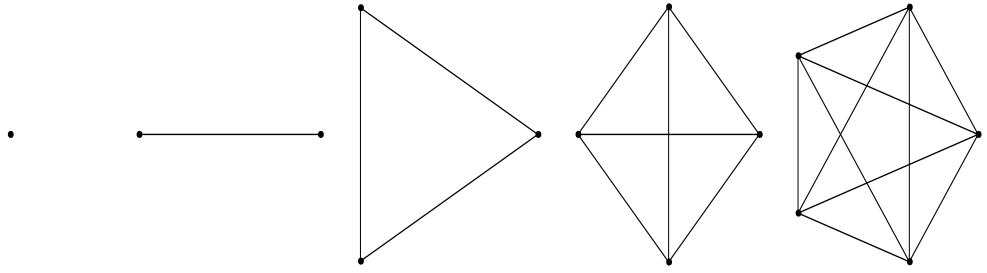
Figure 2.1: Complete Graphs

A graph is **bipartite** if its vertex set can be partitioned into two nonempty subsets X and Y such that each edge of G has one end in X and the other in Y. The bipartite graph G with bipartition (X, Y) is denoted by $G(X,Y)$. A simple bipartite graph $G(X,Y)$ is complete if each vertex of X is adjacent to all the vertices of Y. If $G(X,Y)$ is complete with $|X| = m$ and $|Y| = n$, then $G(X,Y)$ is denoted by $K_{m,n}$. A complete bipartite graph of the form $K_{1,n}$ is called a **star**.

A **walk** in a simple graph G is a sequence $v_0 e_1 v_1 ... v_{k-1} e_n v_n$ of vertices and distinct edges such that consecutive vertices in the sequence are adjacent. The walk is closed if $v_0 = v_n$ and is open otherwise. A **path** is a walk with no repeated vertex. P_n denotes a path on n vertices. The **length** of a walk or path is its number of edges. A **cycle** is a closed walk of length at least three in which the vertices are distinct except the first and the last. C_n denotes a cycle on n vertices. A cycle is odd or even according as its length is odd or even. The graph obtained from C_{n-1} by joining each vertex to a new vertex v is called **wheel**. W_n denotes a wheel on n vertices. A graph is said to be **acyclic** if it has no cycles. A **tree** is a connected acyclic graph.

Theorem 3. *A graph is bipartite if and only if, it contains no odd cycles.*

A graph is **connected** if it has a $u - v$ path for each pair of vertices u and v; otherwise it is **disconnected**. A vertex of a connected graph is a **cut-vertex** if its removal produces a disconnected graph. If e is an edge of a given graph $G = (V, E)$ we use the notation $G - e$ to indicate the graph obtained from G by removing the edge e. Similarly, if v is a vertex of G, we use the notation $G - v$ to indicate the graph obtained from G by removing the vertex v together with all the edges incident to v.

2.3 Matchings and Factors

A **matching** in a graph is a set of edges such that every vertex of the graph is on at most one in the set. Thus, a set of independent edges in G is a matching.

Definition 4. A matching M in a graph G is **maximal** if there is no matching M' in G so that $M \subsetneq M'$.

Definition 5. A matching M in a graph G is **maximum** if there is no matching M' in G so that $|M| < |M'|$.

It is obvious that all maximum matchings are maximal. But the converse is not true.

Definition 6. A **factor** of a graph G is a spanning subgraph of G. A k-factor of G is a factor of G that is k-regular. Thus a 1-factor of G is a matching that includes all the vertices of G. For this reason, a 1-factor of G is called a **perfect matching** of G.

2.4 Eulerian and Hamiltonian Graphs

Euler initiated the study of Graph Theory in 1736 with the famous seven bridges of Königsberg problem. The town of Königsberg straddled the Pregel River with a total of seven bridges connecting the two shores and two islands. The problem was to start from any one of the four land areas, take a stroll across the seven bridges and get back to the starting point without crossing any bridge a second time. This problem can be converted into one concerning the graph obtained by representing each land area by a vertex and each bridge by an edge. Afterwards Euler proved that a tour of all edges in a connected, undirected graph without repetition is possible if and only if the degree of each vertex is even. Such graphs are known as Eulerian Graphs. The Königsberg bridge problem will have a solution provided that this obtained graph is Eulerian. But this is not the case, since it has vertices of odd degrees.

Definition 7. An Eulerian path is a complete tour of all the edges of a graph without repeating any edges. A closed Eulerian path is called an Eulerian cycle. **An Eulerian graph** is a graph that has an Eulerian cycle.

The first theorem of graph theory was due to Leonhard Euler. This theorem connects the degrees of vertices and the number of edges of a graph.

Theorem 8. *(Euler's Theorem) Let G be a connected graph. G is an Eulerian graph if and only if the degree of each vertex is even.*

Theorem 9. *For any graph G which has n vertices and m edges, the sum of the degrees of the vertices equal to twice the number of edges. Thus,*

$$\sum_{i=1}^{n} deg(v_i) = 2m$$

Note that in any graph the sum of all the vertex-degrees is an even number, since each edge contributes exactly 2 to the sum.

Corollary 10. *In a graph, the number of vertices of odd degrees is even .*

Definition 11. A path between two vertices in a graph is a Hamiltonian path if it passes through every vertex of the graph. A closed Hamiltonian path is called a Hamiltonian cycle in the graph. **A Hamiltonian graph** is a graph that has a Hamiltonian cycle.
Theorem 8 provides a simple test for determining whether or not a graph is Eulerian. No such test is known and none is thought to exist to determine whether or not a graph is Hamiltonian.

2.5   NP-Complete Problem

Definition 12. A **decision problem** is a problem that requires only a positive or negative answer regarding whether some element of its domain has a particular property.

Definition 13. A decision problem belongs to the **class P** if there is a polynomial-time algorithm to solve the problem. A decision problem belongs to the **class NP** if there is a way to provide evidence of the correctness of a positive answer so that it can be confirmed by a polynomial-time algorithm.

Definition 14. A problem is said to be **NP-hard** if an algorithm for solving it can be translated into one for solving any other NP-problem. It is much easier to show that a problem is NP than to show that it is NP-hard. A problem which is both NP and NP-hard is called an **NP-complete problem**.

Chapter 3

EDGE COLORINGS

3.1 History of Edge Coloring

The edge-coloring problem is to color all edges of a given graph with the minimum number of colors so that no two adjacent edges are assigned the same color. In this chapter, we historically review the edge-coloring problem which was appeared in 1880 in relation with the four-color problem. The problem is that every map could be colored with four colors so that any neighboring countries have different colors. It took more than 100 years to prove the problem affirmatively in 1976 with the help of computers. The first paper dealing with the edge-coloring problem was written by Tait in 1880. In this paper Tait proved that if the four-color conjecture is true, then the edges of every 3-connected planar graph can be properly colored using only three colors. Several years later, in 1891 Petersen pointed out that there are 3-connected, cubic graphs which are not 3 colorable. The minimum number of colors needed to color edges of G is called the chromatic index $\chi'(G)$ of G. Obviously $\chi'(G) \geq \Delta(G)$, since all edges incident to the same vertex must be assigned different colors. In 1916, König has proved his famous theorem which states that every bipartite graph can be edge-colored with exactly $\Delta(G)$ colors, that is $\chi'(G) = \Delta(G)$. In 1949, Shannon proved that every graph can be edge-colored with at most $\frac{3\Delta(G)}{2}$ colors, that is $\chi'(G) \leq \frac{3\Delta(G)}{2}$. In 1964, Vizing proved that $\chi'(G) \leq \Delta(G) + 1$ for every simple graph. [8]

According to the rapid progress of computers, the research on computer algorithms has become active with emphasis on the efficiency and complexity, and efficient algorithms have been developed for various graph problems. However, Holyer [11] proved that the edge coloring problem is NP-complete, and hence it is very unlikely that there is a polynomial time algorithm for solving the problem. Hence a good approximation algorithm would be useful.

We now give a simple example of an application of the edge coloring. Consider a computer network consisting of several computers connected to some

of the others through communication lines. Suppose that every computer can communicate with at most one other computer in a time unit. We want to make a time schedule such that every computer communicates to all of its neighbors, and to minimize the number of the total necessary time units. Consider this problem by using an edge coloring, as follows. We first regard the network as a graph by replacing each computer with a vertex, and each communication line with an edge. We next find an edge coloring of the resulting graph with the minimum number of colors. Each color represents a time unit at which the corresponding communication line is used. Therefore, the number of used colors is the total number of necessary time units.

Today this example is known as one of the most basic one in the graph coloring theory. A large number of papers dealing with edge coloring have been published. Many of them focus to obtain a small upper bound on the number of necessary colors for colorings, or develop efficient algorithms to solve the coloring problems, that is, to find a coloring such that the number of used colors is as small as possible.

3.2 The Minimization Problem for Edge Coloring

Definition 15. An **edge coloring** of a graph G is a function $f : E(G) \to C$, where C is a set of distinct colors. For any positive integer k, a **k-edge coloring** is an edge coloring that uses exactly k different colors. A **proper edge coloring** of a graph is an edge coloring such that no two adjacent edges are assigned the same color. Thus a proper edge coloring f of G is a function $f : E(G) \to C$ such that $f(e) \neq f(e')$ whenever edges e and e' are adjacent in G.

Definition 16. The **chromatic index** of a graph G, denoted $\chi'(G)$, is the minimum number of different colors required for a proper edge coloring of G. G is **k-edge-chromatic** if $\chi'(G) = k$.

Theorem 17. *For any graph G,*

$$\Delta(G) \leq \chi'(G) \leq 2\Delta(G) - 1$$

Proof. An obvious lower bound for $\chi'(G)$ is the maximum degree $\Delta(G)$ of any vertex in G. This is of course, because the edges incident one vertex must be differently colored. It follows that $\Delta(G) \leq \chi'(G)$. The upper bound can be found by using adjacency of edges. Each edge is adjacent to at most $\Delta(G) - 1$ other edges at each of its endpoints. Thus,

$$1 + (\Delta(G) - 1) + (\Delta(G) - 1) = 2\Delta(G) - 1$$

colors will always suffice for a proper edge coloring of G. $\qquad \square$

Definition 18. The set of all edges receiving the same color in an edge coloring of G is called a **color class**. Alternatively a k-edge coloring can be thought of as a partition $(E_1, E_2, ..., E_k)$ of $E(G)$, where E_i denotes the (possibly empty) subset of $E(G)$ assigned color i. If a coloring $\xi = (E_1, E_2, \cdots, E_k)$ is proper, then each E_i is a matching. Therefore $\chi'(G)$ may be regarded as the smallest number of matchings into which the edge set of G can be partitioned. This interpretation of $\chi'(G)$ will be helpful in the proof of certain useful results.

Theorem 19. *Let G be a graph with m edges and let $m^*(G)$ be the size of a maximum matching. Then,*

$$\chi'(G) \geq \lceil \frac{m}{m^*(G)} \rceil.$$

Proof. Consider coloring of the edges with using $q = \chi'(G)$ colors $\alpha_1, \alpha_2, \cdots, \alpha_q$ and let E_i denote the set of edges with color α_i. We have

$$m = |E_1| + |E_2| + \cdots + |E_q| \leq qm^*(G)$$

Hence, $q \geq \frac{m}{m^*(G)}$ and $\chi'(G) \geq \lceil \frac{m}{m^*(G)} \rceil$. $\qquad\square$

3.3 Line Graphs

Definition 20. $L(G)$ is a graph whose vertices are the edges of G, such that edges with a common endpoint in G are adjacent in $L(G)$. This graph, $L(G)$ is called the **line graph** of G.

Some simple properties of the line graph $L(G)$ of a graph G are presented as follows:

1. G is connected if and only if $L(G)$ is connected.

2. If H is a subgraph of G, then $L(H)$ is a subgraph of $L(G)$.

3. If e is an edge of G joining u and v, then the degree of e in $L(G)$ is the same as the number of edges of G adjacent to e in G.
   Hence $deg(e) = deg(u) + deg(v) - 2$.

4. It is relatively easy to determine the number of vertices and edges of the line graph $L(G)$ in terms of quantities in G. The order of $L(G)$ is equal to the size of G. In counting the number of edges in $L(G)$, we have to examine only the vertices in G, with the degree more than 1. Each edge of $L(G)$ corresponds to a pair of adjacent edges in G. Let $v = \{1, 2, \cdots, n\}$ be the vertex set of G, and let d_i be the degree of vertex i. If $d_i > 1$, any two of the d_i edges that are incident at vertex i can be chosen in $C(d_i, 2)$ ways. Two edges e and e' of G that are incident at vertex i correspond to two vertices in $L(G)$ joined by an edge. Let the degrees of the vertices of G be $d_1, d_2, ..., d_n$. Thus the total number of edges in $L(G)$ is:

$$
\begin{aligned}
m(L(G)) &= \sum_{i=1}^{n} \binom{d_i}{2} \\
&= \frac{d_1!}{2!(d_1 - 2)!} + \frac{d_2!}{2!(d_2 - 2)!} + ... + \frac{d_n!}{2!(d_n - 2)!} \\
&= \frac{d_1(d_1 - 1)}{2} + \frac{d_2(d_2 - 1)}{2} + ... + \frac{d_n(d_n - 1)}{2} \\
&= \frac{[d_1^2 + d_2^2 + ... + d_n^2] - [d_1 + d_2 + ... + d_n]}{2} \\
&= \frac{1}{2}[\sum_{i=1}^{n} d_i^2] - m(G)
\end{aligned}
$$

Theorem 21. *The line graph of the star $K_{1,n}$ is the complete graph K_n.*

Theorem 22. *The line graph of cycle is a cycle.*

Theorem 23. *The line graph of a graph G is a path if and only if G is a path.*

Proof. Let G be the path P_n on n vertices. Then clearly $L(G)$ is the path P_{n-1} on $n - 1$ vertices.

Conversely, let $L(G)$ be a path. Then no vertex of G can have degree greater than 2. Because, if G has a vertex v of degree greater than 2, the edges incident to v would form a complete subgraph of $L(G)$ with at least three vertices. Hence G must be either a cycle or a path. But G cannot be a cycle, because the line graph of a cycle is a cycle. □

A line graph can be used to convert an edge coloring problem into a vertex coloring problem. This observation appears to be of little value in computing chromatic indices, however, since chromatic numbers are extremely difficult to evaluate in general [2].

Theorem 24. *The chromatic index of a graph G equals the chromatic number of its line graph $L(G)$.*

Proof. From the definitions it is immediate that $\chi'(G) = \chi(L(G))$, where $L(G)$ is the line graph of G. □

11

3.4 Chromatic Index for Common Graph Families

Proposition 25. *Complete Graphs:*

$$\chi'(K_n) = \begin{cases} n - 1, & \text{if } n \text{ is even;} \\ n, & \text{if } n \text{ is odd.} \end{cases}$$

Proof. Since K_n is regular of degree $n - 1$, $\quad \chi'(K_n) \geq n - 1$.

Case 1: n is even. We now show that $\chi'(K_n) \leq n - 1$ by exhibiting a proper $(n - 1)$-edge coloring of K_n. Label the n vertices of K_n as $0, 1, ..., n - 1$. Draw a circle with center at 0 and place the remaining $n-1$ numbers on the circumference of the circle so that they form a regular $(n - 1)$-gon (see Figure 3.1). Then the $\frac{n}{2}$ edges $(0, 1), (2, n - 1), (3, n - 2), \cdots, (\frac{n}{2}, \frac{n}{2} + 1)$ form a 1-factor of K_n. These $\frac{n}{2}$ edges are thick edges of Figure 3.1. Rotation of these edges through the angle $\frac{2\pi}{2n-1}$ in succession gives $(n-1)$-edge disjoint 1-factors of K_n. This would account for $\frac{n}{2}(n - 1)$ edges, or all the edges of K_n. Each 1-factor can be assigned a color. Thus $\chi'(K_n) \leq n - 1$. This proves the result in Case 1.
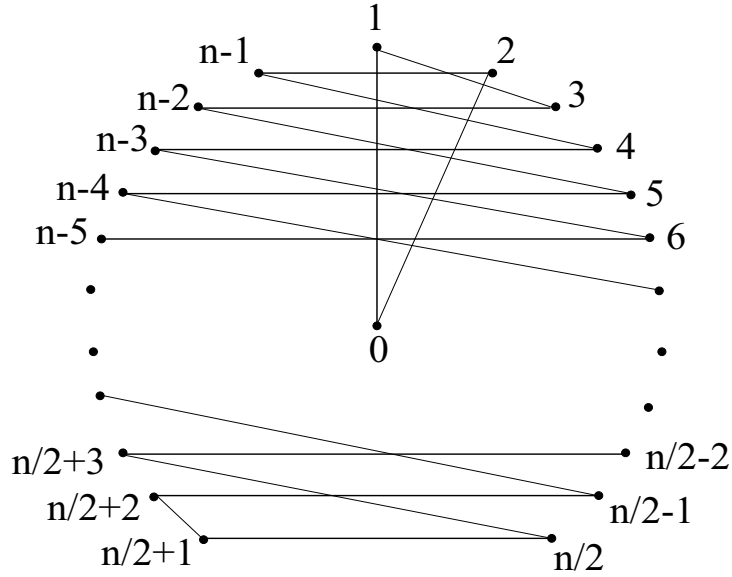
Figure 3.1: Graph for proof of Proposition 25.

Case 2: n is odd. Take a new vertex and make it adjacent to all the n vertices of K_n. This gives K_{n+1}. By Case 1, $\chi'(K_{n+1}) = n$. The restriction of this edge coloring to K_n yields a proper n-edge coloring of K_n. Hence $\chi'(K_n) \leq n$. However, K_n cannot be edge colored properly with $n - 1$ colors. This is because the size of any matching of K_n can contain no more than $\frac{n-1}{2}$ edges and hence $n-1$ matchings of K_n can contain no more than $\frac{(n-1)^2}{2}$ edges. But K_n has $\frac{n(n-1)}{2}$ edges. Thus, $\chi'(K_n) \geq n$, and hence $\chi'(K_n) = n$. $\qquad\square$

Proposition 26. *Path Graphs:*

$$\chi'(P_n) = 2, \ for \ n \geq 3.$$

Proposition 27. *Cycle Graphs:*

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

Proposition 28. *Trees:*

$$\chi'(T) = \Delta(T), \ for \ any \ tree \ T.$$
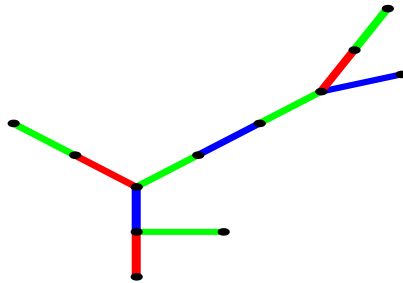
Figure 3.2: A proper edge 3-coloring of a tree.

Proposition 29. *Wheel Graphs:*

$$\chi'(W_n) = n - 1, \ for \ n \geq 4.$$

Figure 3.3: A proper edge 6-coloring of a wheel.
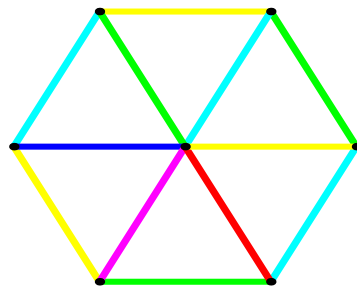
A 3-regular graph is also called a **cubic graph**. The best known cubic graph is the Petersen Graph (see Figure 3.4). The Petersen Graph is 3-regular with chromatic index 4. It is also not Hamiltonian. We will see now that these properties are connected.
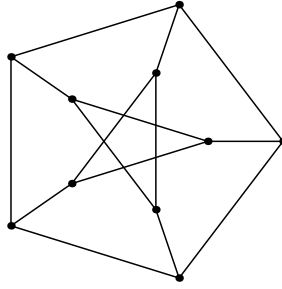
13

Figure 3.4: Petersen Graph.

Theorem 30. *Let G be a 3-regular graph with chromatic index 4. Then G is not Hamiltonian.*

Proof. Since G is 3-regular then it must have an even number of vertices. Suppose G is Hamiltonian, then any Hamiltonian cycle of G is even, so we can color its edges properly with 2 colors, say red and blue. Now each vertex is incident with 1 red edge, 1 blue edge and 1 uncolored edge. The uncolored edges form a 1-factor of G, so we can color all of them with the same color, say green. Thus, G must be 3-edge-colorable, which is impossible. Therefore, G cannot be Hamiltonian. $\square$

3.5 Chromatic Incidence

The next few definitions pertain to all edge-colorings, not just to proper ones [9].

Definition 31. For a given edge coloring of a graph, **color i is incident** on vertex v if some edge incident on v has been assigned color i. Otherwise, **color i is missing** at vertex v.

Definition 32. The **chromatic incidence** at v of a given edge coloring f is the number of different edge-colors incident on vertex v. It is denoted by $ecr_v(f)$.

Definition 33. The **total chromatic incidence** for an edge coloring f of a graph G, denoted by $ecr(f)$, is the sum of the chromatic incidences of all the vertices. That is,

$$ecr(f) = \sum_{v \in V_G} ecr_v(f)$$

The following four statement are immediate consequences of the definitions.

Proposition 34. *Let f be any edge coloring of a graph G. Then for every $v \in V_G$,*

$$ecr_v(f) \leq deg(v)$$

Corollary 35. *Let f be any edge coloring of a graph G. Then*

$$\sum_{v \in V(G)} ecr_v(f) \leq \sum_{v \in V(G)} deg(v)$$

14

Proposition 36. *An edge coloring f of a graph G is proper if and only if for every vertex $v \in V(G)$*

$$ecr_v(f) = deg(v)$$

Corollary 37. *An edge coloring f of a graph G is proper if and only if*

$$\sum_{v \in V(G)} ecr_v(f) = \sum_{v \in V(G)} deg(v)$$

3.6   Edge Coloring of Bipartite Graphs

The edge coloring problem for bipartite graphs can be used to model a time table problem: assume that, given a set of teachers and a set of classes, it is known which classes and how many hours each teacher must teach, then construct a time table to minimize the schooltime. Represent teachers and classes as vertices of a graph. When a teacher must teach a class h hours, join the vertices corresponding to the teacher and the class by h multiple edges. The edges colored with a same color correspond to classes that can be held simultaneously. Thus the timetable to minimize the schooltime corresponds to an edge coloring of the graph with the minimum number of colors.

Definition 38. In a graph G with an (possibly improper) edge coloring, a **Kempe i-j edge-chain** is a component of the subgraph of G induced on all the i-colored and j-colored edges.

The following two lemmas establish facts about the chromatic degree that are used for the characterization of bipartite graphs. They involve edge colorings that are not assumed to be proper. The first lemma makes use of the properties of an Eulerian graph.

Lemma 39. *Let G be a connected graph with at least two edges that is not an odd cycle graph. Then G has an edge 2-coloring such that both colors are incident on every vertex of degree at least 2.*

Proof. **Case 1:** G is Eulerian.
If G is a cycle, then by the hypothesis of the theorem it must be of even length. In such a case it is easy to verify that G has an edge 2-coloring having the required property. If G is not a cycle, then it must have a vertex v_0 of degree at least 4. Let $v_0, e_1, v_1, e_2, v_2, e_3, \cdots, e_m, v_0$ be an Euler Cycle and let

$$E_1 = \{e_i \mid i \ is \ odd\} \quad and \quad E_2 = \{e_i \mid i \ is \ even \} \tag{3.1}$$

Then (E_1, E_2) is an edge 2-coloring in which at every vertex both colors are represented because in the Euler Cycle considered every vertex including v_0 appears as an internal vertex.

Case 2: G is not Eulerian.

In this case G must have an even number of odd vertices. Now construct an Eulerian Graph G' by adding a new vertex v_0 and connecting it to every vertex of odd degree in G. Let $v_0, e_1, v_1, e_2, v_2, e_3, \cdots, e_m, v_0$ be an Eulerian Cycle in G'. If E_1 and E_2 are defined as in 3.1, then (E_1, E_2) is an edge 2-coloring of G' in which both colors are represented at each vertex of G. It may be now be verified that $(E_1 \bigcap E, E_2 \bigcap E)$ is an edge 2-coloring of G in which both colors are represented at each vertex of degree at least 2.

$\square$

Lemma 40. *Let f be an edge k-coloring of a graph G with the largest possible total chromatic incidence. Let v be a vertex on which some color i is incident at least twice and on which some color j is not incident at all. Then the Kempe i-j edge-chain K containing vertex v is an odd cycle.*

Proof. By Lemma 39, if the Kempe i-j edge-chain K incident on vertex v were not an odd cycle, then we could rearrange the edge colors i and j within K so that the chromatic incidence of the coloring of K would be 2 at every vertex. The edge coloring for G thereby obtained would have higher chromatic incidence at vertex v at least equal chromatic incidence at every other vertex of G. This would contradict the premise that the edge coloring f has the maximum possible total chromatic incidence. $\square$

In 1916, König [12], while studying the factorization of the determinants of matrices, proved his famous theorem which can be stated as follows:

Theorem 41. *(König's Theorem)   Let G be a bipartite graph. Then,*
$\chi'(G) = \Delta(G)$.

Proof. By the way of contradiction, suppose that $\chi'(G) \neq \Delta(G)$. Then by Theorem 17, $\Delta(G) < \chi'(G)$. Next, let f be an edge $\Delta(G)$-coloring of graph G for which the total chromatic incidence $ecr_G(f)$ is maximum. Since f is not a proper edge coloring, by Proposition 34 there is a vertex v such that $ecr_v(f) < deg(v)$. Thus some color occurs on at least two edges incident on v. But there are $\Delta(G) - 1$ other colors and at most $\Delta(G) - 2$ other edges incident on v, which means that some other color is not incident on vertex v. It follows by Lemma 40, that graph G contains an odd cycle, which contradicts the fact that G is bipartite. $\square$

3.6.1   Edge Colorings of Regular Bipartite Graphs

König's edge coloring theorem is directly equivalent to the special case of regular bipartite graphs (since any bipartite graph of maximum degree $\Delta(G)$ is a subgraph of a $\Delta(G)$-regular bipartite graph). Rizzi [17] gave the following very elegant short argument for the k-edge-colorability of k-regular bipartite graphs.

Let G be a counterexample with fewest edges. So G has no perfect matching. Choose an edge $e = uv$. Then we can extend the graph $G - u - v$ to a k-regular bipartite graph H by adding at most $k - 1$ new edges. As H has fewer edges than G, H has a k-edge coloring. Since less than k new edges have been added, there is a color M that uses none of the new edges. Then $M \cup \{e\}$ is a perfect matching in G, a contradiction.

3.7 Vizing's Theorem

Definition 42. Let G be a graph, and let f be a proper edge k-coloring of a subset S of the edges of G. Then f is **blocked** if for each uncolored edge e, every color has already been assigned to the edges that are adjacent to e. Thus, f cannot be extended to any edge outside subset S.

Lemma 43. *Let i and j be two of the colors used in a proper edge coloring of a graph G. Then every Kempe i-j edge-chain K is a path (open or closed).*

Proof. Every vertex of Kempe chain K has degree at most 2 (since the edge coloring is proper), and by Definition 38, K is a connected subgraph. □

Theorem 44. *(Vizing's Theorem) Let G be a simple graph. Then there exists a proper edge coloring of G that uses at most $\Delta(G) + 1$ colors.*

Proof. To construct such an edge coloring, start by successively coloring edges using any method until the coloring is blocked or complete. If the set of uncolored edges is empty, then the construction is complete. Otherwise, there is some edge e with endpoints u and v that remains uncolored. It will be shown that by recoloring some edges, the blocked coloring can be transformed into one that can be extended to edge e. The process can then be repeated until all edges have been colored.

Since the number of colors exceeds $\Delta(G)$, it follows that at each vertex at least one of the colors is missing. Let c_0 be a color missing at vertex u, and c_1 a color missing at vertex v. Color c_1 cannot also be missing at vertex u, since if it were, edge e would not have remained uncolored. (For the same reason, color c_0 must occur at vertex v.)

So let e_1 be the c_1-edge-incident on vertex u, and let v_1 be its other endpoint. Next, let c_2 be a color missing at v_1. If c_2 is also missing at vertex u, then the color of the edge e_1 can be changed from c_1 to c_2, thereby permitting the assignment of color c_1 to edge e, as illustrated in Figure 3.5.

If color c_2 does occur at vertex u, then let e_2 be the c_2-edge incident on vertex u, let v_2 be its other endpoint, and let c_3 be a color missing at vertex v_2. Continue iteratively in this way so that at the j th iteration, e_j is the c_j-edge incident on vertex u, v_j is its other endpoint, and c_{j+1} is the color missing at vertex v_j. Let m be the smallest j such that vertex v_m has a missing color c_{m+1} such that c_{m+1}
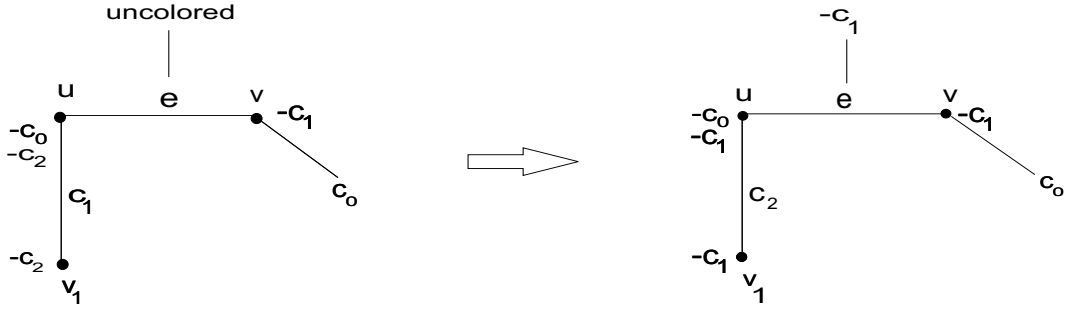
Figure 3.5: Extending an edge coloring to edge e by recoloring edge c_1.

is also missing at vertex u or is one of the colors in the list $c_1, c_2, ..., c_m$. (Such an m exits, since the set of colors is finite.)

Case 1: *Color c_{m+1} is missing at both vertex v_m and vertex u. (Color Shift)*

Then perform the following color shift: for $j = 1, ..., m$ change the color of edge e_j from c_j to c_{j+1}. This releases color c_1 from edge e_1 so that it can be reassigned to edge e. The color shift is illustrated in Figure 3.6. Notice that it maintains a proper edge coloring because, by the construction, color c_{j+1} was missing at both endpoints of edge e_j before the shift.
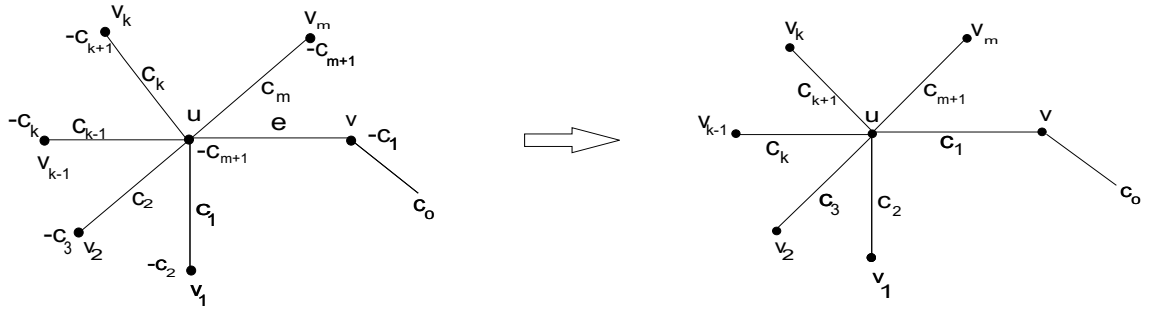
Figure 3.6: **Case 1:** Color shift to free color c_1 for edge e.

Case 2: *Color $c_{m+1} = c_k$, where $1 \leq k \leq m$. (Swap and Shift)*

Let K be the Kempe $c_0 - c_k$ edge-chain incident on vertex v_m. By definition of m, K includes the c_0-edge incident on v_m, but there is no c_k-edge incident on vertex v_m. By Lemma 43, Kempe chain K is a path, and one end of this path is vertex v_m. There are three subcases to consider, according to where the other end of the path is. In each of the three subcases, the two colors are swapped so that a Case 1 color shift can then be performed.

Case 2a: *Path K reaches vertex v_k.*

Then swap colors c_0 and c_k along path K. As a result of the swap, color c_k no longer occurs at vertex u. This configuration permits a Case 1 color shift that releases color c_1 for edge e. The swap and shift are illustrated in Figure 3.7.
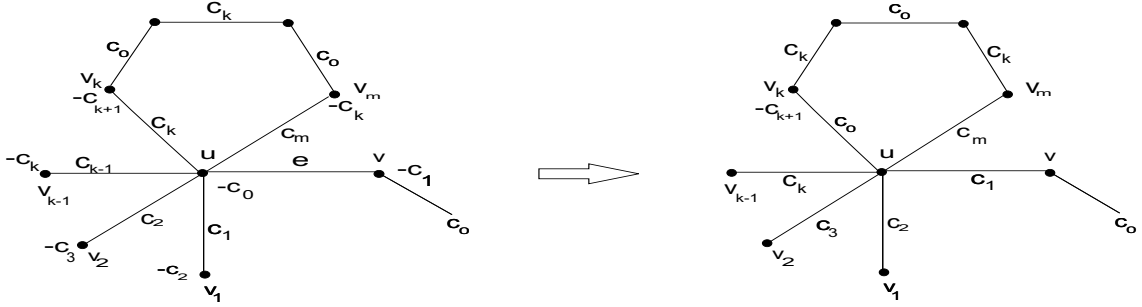
Figure 3.7: **Case 2a:** Swap and shift.

Case 2b: *Path K reaches the vertex v_{k-1}.*

Then swap colors c_0 and c_k along path K. As a result of the swap, color c_0 no longer occurs at vertex v_{k-1}. Thus, edge e_{k-1} can be recolored c_0, as in Figure 3.8, so that color c_{k-1} no longer occurs at vertex u. A Case 1 shift can now be performed to release color c_1 for edge e.
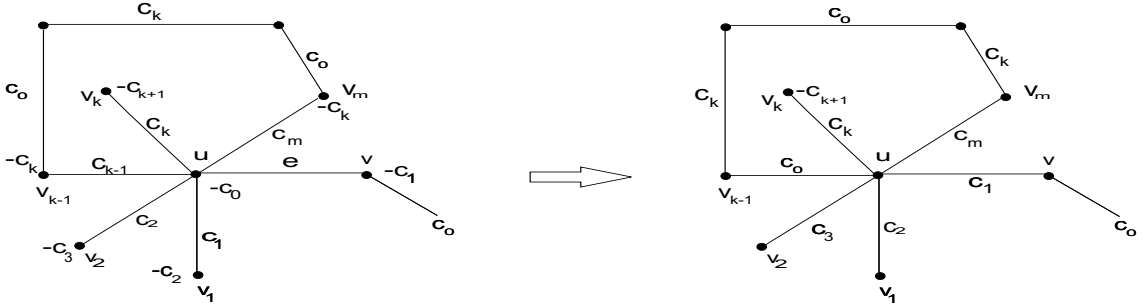
Figure 3.8: **Case 2b:** Swap, recolor edge e_1, and then shift.

Case 2c: *Path K never reaches vertex v_{k-1} or vertex v_k.*

Since color c_0 does not occur at vertex u, and since color c_k occurs at u only on the edge from v_k, it follows that path K does not reach vertex u. Then swap colors c_0 and c_k along path K, so that color c_0 no longer occurs at vertex v_m. Now perform a Case 1 color shift that releases color c_1 for edge e, as in Figure 3.9.
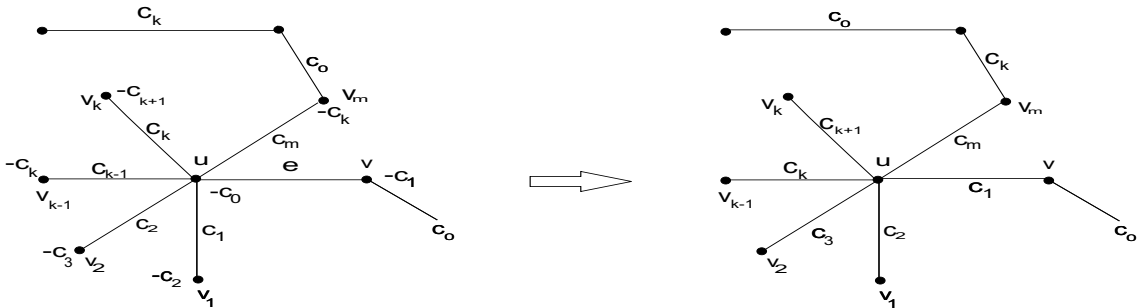
Figure 3.9: **Case 2c:** Swap and shift.

$\square$

19

Corollary 45. *Let G be a simple graph. Then either $\chi'(G) = \Delta(G)$ or $\chi'(G) = \Delta(G) + 1$.*

3.7.1 The Classification Problem

Vizing's Theorem for simple graphs gives us a basic way of classifying graphs into two classes. A graph G is said to be of **Class 1** if $\chi'(G) = \Delta(G)$, and of **Class 2** if $\chi'(G) = \Delta(G) + 1$. We have already seen that even complete graphs, even cycle graphs, and bipartite graphs are of Class 1, and that odd cycle graphs and odd complete graphs are of Class 2. More generally, every regular graph of odd order is of Class 2. It is not true, however, that every regular graph of even order is of Class 1; the Petersen Graph, for example, is Class 2. However the general problem of deciding which graphs belong to which class is unsolved.

It seems natural to expect that the more edges a graph has, the more likely it is to be of Class 2. This idea is made precise in the following result, which gives a sufficient condition for a graph to be of Class 2. This elementary result was proved by Beineke and Wilson [3].

Theorem 46. *Let G be a graph with n vertices and m edges. If*

$$m > \Delta(G) \cdot \lfloor \frac{1}{2}n \rfloor,$$

then G is of Class 2.

Proof. Let G be a graph with n vertices and m edges and $m > \Delta(G) \cdot \lfloor \frac{1}{2}n \rfloor$. Assume that G is of Class 1. Then $\chi'(G) = \Delta(G)$. Let a $\Delta(G)$-edge coloring of G be given. Then any $\Delta(G)$-coloring of the edges of G partitions the set of edges into $\Delta(G)$ independent subsets. But the number of edges in each independent subset cannot exceed $\lfloor \frac{1}{2}n \rfloor$, since otherwise two of these edges would be adjacent. It follows that $m \leq \Delta(G) \cdot \lfloor \frac{1}{2}n \rfloor$, giving the required contradiction. $\square$

Definition 47. A graph G is called **overfull** if $m > \Delta(G) \lfloor \frac{1}{2}n \rfloor$.

Corollary 48. *Every overfull graph is of Class 2.*

Proof. By the definition of overfull graph and Theorem 46, it is obvious. $\square$

Hilton [10] conjectured that a graph G of order n with $\Delta(G) > \frac{n}{3}$ is of Class 2 if and only if G contains an overfull subgraph H with $\Delta(G) = \Delta(H)$.

Corollary 49. *Every regular graph of odd order is of Class 2.*

Corollary 50. *If H is a regular graph of even order, and if G is any graph obtained from H by inserting a new vertex into any edge of H, then G is of Class 2.*

We can also deduce the following result of Vizing:

Corollary 51. *If G is a regular graph containing a cut-vertex, then G is of Class 2.*

Proof. If G is of odd order, then the result follows from Corollary 49. If G is of even order, let $G = H \cup K$, where $H \cap K = \{v\}$. We may assume that H has odd order (say k), and that every vertex of H has degree $\Delta(G)$, except for v whose degree in H is less than $\Delta(G)$. It follows that the number of edges of H is:

$$m(H) = \frac{1}{2}[(k-1)\Delta(G) + deg_H(v)] > \Delta(G)\lfloor \frac{1}{2}k \rfloor$$

and the result follows from Theorem 46. $\square$

3.7.2 Vizing's Adjacency Lemma

A graph G with at least two edges is **minimal with respect to chromatic index** if $\chi'(G - e) = \chi'(G) - 1$ for every edge e of G. Since isolated vertices have no effect on edge colorings, it is natural to rule out isolated vertices when considering such minimal graphs. Therefore, the added hypothesis is that a minimal graph G is connected is equivalent to the assumption that G has no isolated vertices.

Two of the most useful results dealing with these minimal graphs are also results of Vizing [22], which are presented without proof.

Theorem 52. *Let G be a connected graph of Class 2 that is minimal with respect to chromatic index. Then every vertex of G is adjacent to at least two vertices of degree $\Delta(G)$. In particular, G contains at least three vertices of degree $\Delta(G)$.*

Theorem 53. *Let G be a connected graph of Class 2 that is minimal with respect to chromatic index. If u and v are adjacent vertices with $deg(u) = k$, then v is adjacent to at least $\Delta(G) - k + 1$ vertices of degree $\Delta(G)$.*

We next examine to which class a graph belongs if it is minimal with respect to chromatic index.

Theorem 54. *Let G be a connected graph with $\Delta(G) = d \geq 2$. Then G is minimal with respect to chromatic index if and only if either:*
i) G is of Class 1 and $G = K_{1,d}$ or
ii) G is of Class 2 and $G - e$ is of Class 1 for every edge e of G.

Proof. Assume first that $G = K_{1,d}$. Then, $\chi'(G) = \Delta(G) \geq 2$ while $\chi'(G - e) = \Delta(G) - 1$ for every edge e of G. Since G is of Class 1, then $\chi'(G - e) = \chi'(G) - 1$.

Next, suppose that G is of Class 2 and $G - e$ is of Class 1 for every edge e of G. Then for an arbitrary edge e of G, we have

$$\chi'(G - e) = \Delta(G - e) < 1 + \Delta(G) = \chi'(G)$$

$$\Rightarrow \chi'(G - e) < \chi'(G).$$

Therefore $\chi'(G - e) + 1 = \chi'(G)$, so G is minimal.

Assume that $\chi'(G - e) < \chi'(G)$ for every edge e of G.

If G is Class 1, then,

$$\Delta(G) \leq \Delta(G - e) + 1 \leq \chi'(G - e) + 1 = \chi'(G) = \Delta(G).$$

Therefore, $\Delta(G-e) = \Delta(G)-1$ for every edge e of G which implies that $G = K_{1,d}$. If G is of Class 2, then

$$\chi'(G - e) + 1 = \chi'(G) = \Delta(G) + 1$$

so that $\chi'(G - e) = \Delta(G)$ for every edge e of G. Suppose that G contains an edge e_1 such that $G - e_1$ is of Class 2. Then

$$\chi'(G - e_1) = \Delta(G - e_1) + 1$$

$\Rightarrow \Delta(G) = \Delta(G-e_1)+1 \Rightarrow \Delta(G-e_1) < \Delta(G)$, implying that G has at most two vertices of degree $\Delta(G)$. This however, contradicts Theorem 52 and completes the proof.

$\square$

A graph G with at least two edges is called **class minimal** if G is of Class 2 and $G - e$ is of Class 1 for every edge e of G. It follows that a class minimal graph without isolated vertices is necessarily connected. On the basis of the Theorem 54, we conclude that except for star graphs, class minimal graphs are connected graphs that are minimal with respect to chromatic index, and conversely.

A lower bound on the size of class minimal graphs is given next in yet another result by Vizing [22].

Theorem 55. *If G is a class minimal graph of size m with $\Delta(G) = d$, then*

$$m \geq \frac{1}{8}(3d^2 + 6d - 1)$$

Proof. Without loss of generality, we assume that G is connected. Suppose that $\delta(G) = k$ and $deg(u) = k$. By Theorem 52, the vertex u is adjacent to at least two vertices of degree d. Let v be such a vertex. By Theorem 53, v is adjacent to at least $d - k + 1$ vertices of degree d. Since the order of G is at least $d + 1$, we arrive at the following lower bound on the sum of degrees of G:

$$2m \geq [d(d - k + 2) + k(k - 1)] = k^2 - k(d + 1) + (d^2 + 2d) \qquad (3.2)$$

However, expression 3.2 is minimized when $k = \dfrac{d + 1}{2}$ so that,

$$2m \geq (\frac{d + 1}{2})^2 - \frac{(d + 1)^2}{2} + d^2 + 2d$$

or

$$m \geq \frac{1}{8}(3d^2 + 6d - 1).$$

$\square$

3.8 Chromatic Index for Multigraphs

Vizing also obtained a corresponding bound for multi-graphs, which is sometimes better than that given by Shannon [23]. It involves the **maximum multiplicity** μ of a multi-graph M, defined to be the maximum number of edges joining any pair of vertices in M. This result is stated as follows:

Theorem 56. *If M is a multi-graph with maximum degree Δ and maximum multiplicity μ, then $\Delta(M) \leq \chi'(M) \leq \Delta(M) + \mu$.*

We conclude this section by showing how Vizing's Theorem for multi-graphs can be used to prove Shannon's result [18]:

Theorem 57. *(Shannon's Theorem) If M is a multi-graph with maximum degree $\Delta(M)$, then*

$$\chi'(M) \leq \frac{3}{2}\Delta(M).$$

Proof. Let M be a multi-graph for which $\chi'(M) = k$, where $k > \frac{3}{2}\Delta(M)$. We may assume that $\chi'(M - e) = k - 1$, for each edge e of M. It follows from the previous theorem that $k \leq \Delta(M) + \mu$, where μ is the maximum multiplicity of M and so there must exist vertices v and w which are joined by at least $k - \Delta(M)$ edges. We now color all of the edges of M except one of the edges joining v and w; since $\chi'(M - e) = k - 1$, this coloring can be done with $k - 1$ colors. Now the number of colors missing from v and w (or both) can not exceed $(k - 1) - (\mu - 1)$, which in turn can not exceed $\Delta(M)$, since $k \leq \Delta(M) + \mu$. But the number of colors missing from

v is at least $(k-1)-(\Delta(M)-1) = k-\Delta(M)$ and similarly the number of colors missing from w is at least $k-\Delta(M)$. It follows that the number of colors missing from both v and w is at least $(k-\Delta(M))+(k-\Delta(M)) = (2k-\Delta(M))-\Delta(M)$, which is positive since $k > \frac{3}{2}\Delta(M)$. By assigning one of these missing colors to the uncolored edge joining v and w, we have colored all of the edges of M using only $k-1$ colors, thereby contradicting the fact that $\chi'(M) = k$. This contradiction establishes the theorem. $\qquad\square$

3.9 Edge Colorings of Planar Graphs

Definition 58. A **planar graph** is a graph which can be embedded in the plane so that no two edges intersect geometrically except at a vertex to which they are both incident.

In this section we briefly consider edge colorings of planar graphs here. Our main problem remains to determine which planar graphs are of Class 1 and which are of Class 2.

Proposition 59. *If G is a planar graph whose maximum degree is at most 5, then G can lie in either Class 1 or Class 2.*

It is easy to find planar graphs G of Class 1 for which $\Delta(G) = d$ for each $d \geq 2$ since all star graphs are planar and of Class 1. There exist planar graphs G of Class 2 with $\Delta(G) = d$ for $d = 2, 3, 4, 5$. For $d = 2$, the graph K_3 has the desired properties. It is not known whether there exists planar graphs of Class 2 having maximum degree 6 or 7; however Vizing [24] proved that if G is planar and $\Delta(G) \geq 8$, then G must be of Class 1. We shall prove a similar, but weaker, result which may be found in his earlier paper.

Theorem 60. *If G is a planar graph with $\Delta(G) \geq 10$, then G is of Class 1.*

Proof. Suppose that the theorem is not true, and suppose G is a planar graph of Class 2 with $\Delta(G) \geq 10$. Without loss of generality, that G is minimal with respect to chromatic index. Since G is planar, there must be at least one vertex in G whose degree is at most 5. Let S denote the set of all such vertices. Define $H = G - S$. Since H is planar, H contains a vertex w such that $deg_H(w) \leq 5$. Because $deg_G(w) > 5$, the vertex w is adjacent to vertices of S. Let $v \in S$ such that $wv \in E(G)$, and let $deg_G(v) = k \leq 5$. Then by Theorem 53, w is adjacent to at least $d-k+1$ vertices of degree d, but $d-k+1 \geq 6$ so that w is adjacent to at least six vertices of degree d. Since $d \geq 10$, w is adjacent to at least six vertices of H, contradicting the fact that $deg_H(w) \leq 5$. $\qquad\square$

As we mentioned above, this result can be improved to show that every planar graph with $\Delta(G) \geq 8$ is of Class 1. However the problem of determining

what happens when the maximum degree is either 6 or 7 remains open. In this connection, the following conjecture was formulated by Vizing [24]:

Planar Graph Conjecture: Every planar graph with maximum degree 6 or 7 is of Class 1.

Seymour [6] conjectured that a planar graph is of Class 2 if and only if G contains an overfull subgraph H with $\Delta(G) = \Delta(H)$. If true, this conjecture would imply that every planar graph G with $\Delta(G) \geq 6$ is of Class 1.

3.10 Edge Game Coloring of Graphs

The notion of a graph coloring game was first introduced by Bodlaender in 1991 [4]. Consider a two person game as follows, played on an uncolored graph $G = (V, E)$ with respect to a set $C = \{1, 2, \cdots, k\}$ of colors. Two players, Player 1 and Player 2, move alternately with Player 1 moving first. Each feasible move consists of choosing an uncolored edge, and coloring it with a color from C, so that in the subgraph H of G induced by the colored edges, incident edges get distinct colors. The game ends when no more feasible move is possible. In this edge coloring game, Player 2 wins if, any stage of the game, there is an uncolored edge adjacent to colored edges in all k colors; otherwise Player 1 wins. So Player 2 tries to surround an uncolored edge with a high number of differently colored edges. Player 1 tries to prevent this [13].

A graph G is called k**-edge-game colorable** if Player 1 has a winning strategy with k colors, and the **game chromatic index** $\chi'_g(G)$ is the smallest number k such that G is k-edge-game colorable.

For each graph G, the edge set $E(G)$ can be partitioned into $k = \chi'(G)$ matchings, $E_1, E_2, \cdots, E_k$. We have called each of them as a color class of G. It is obvious that in the edge game coloring of graph G, Player 1 always want to color the edges in a color class with the same color, and Player 2 try to color each edge in a matching with distinct colors.

To facilitate the study of game chromatic index, we consider the following edge game coloring on a graph G. Let us consider the game chromatic index of K_5. It is known that $\chi'(K_5)$ and each matching of K_5 contains at most two edges. Let $C = 1, 2, \cdots, 6$. Initially, Player 1 color an arbitrary edge. Suppose Player 2 has just colored an edge e with color $i \in C$. If there is still uncolored edge, then Player 1 choose an edge e' which has no common vertex with e and color it with i also. This guarantees that there are 8 edges of G are colored with 4 colors of C. Therefore, Player 1 has a winning strategy using 6 colors. Thus

$\chi'(K_5) = 6$.

An obvious lower bound of $\chi'_g(G)$ is $\chi'(G)$. This lower bound can be reached because $\chi'_g(K_{1,r}) = \chi'(K_{1,r}) = r$, and $\chi'_g(G)$ can be strictly greater than $\chi'(G)$ because $\chi'_g(P_n) = 3$ but $\chi'(P_n) = 2$ when $n \geq 5$. We can also see that $\chi'_g(C) = 3$.

Note that for any graph G of maximum degree $\Delta(G)$, we have $\Delta(G) \leq \chi'_g(G) \leq 2\Delta(G) - 1$ since no edge is adjacent to more than $2\Delta(G) - 2$ edges. This motivates us to consider graphs whose chromatic game indices are bounded above by $\Delta(G) + c$ for some constant c.

The game chromatic index has first been studied by Lam, Shiu and Xu [13], who show that trees with maximum degree $\Delta(G)$ have game chromatic index at most $\Delta(G) + 2$ and mention that the class of trees with maximum degree 3 has game chromatic at most $\Delta(G) + 1 = 4$. Then, in 2002, Erdös, Faigle, Hochstättler and Kern [7] proved that the statement for $\Delta(G) \geq 6$. Now only the cases $\Delta(G) = 4$ and $\Delta(G) = 5$ are open.

Theorem 61. $\chi'_g(T) \leq \Delta(T) + 1$, for any tree T with $\Delta(T) \geq 6$.

The bound in Theorem 61 is easily seen to be sharp:

Theorem 62. For any $\Delta(T) \geq 2$ there exists a tree with the game chromatic index equal to $\Delta(T) + 1$.

The game chromatic index of wheels also shown by Lam, Shiu and Xu [13]. It is clear that $\chi'_g(W_n) \leq n + 2$.

Theorem 63. $\chi'_g(W_4) = 5$ and $\chi'_g(W_n) = n + 1$ when $n \geq 4$.

Chapter 4

EDGE COLORING WITH web*MATHEMATICA*

In this chapter, we present the problem of edge coloring of graphs with web*Mathematica* by using the Combinatorica package. First, we introduce in Section 4.1 the package of Combinatorica. In Section 4.2, we describe the concept of web*Mathematica* which allows the generation of dynamic web content with Mathematica. In Section 4.3 and 4.4, we give examples of drawing graphs, and their chromatic indices by using webMathematica. The URL adress of this page is http://gauss.iyte.edu.tr:8080/webMathematica/atina/edge.jsp.

4.1 Mathematica and Combinatorica Package

Mathematica is a high-level programming language, a calculator, a composite of mathematical algorithms, and a program that is more powerful than any application that has been devised for it. It is also a system developed recently for doing mathematics by computer. Combinatorica, an extension to the popular computer algebra system Mathematica, is the most comprehensive software available for educational and research applications of discrete mathematics, particularly combinatorics and graph theory. It includes functions for constructing graphs and other combinatorial objects, computing invariants of these objects, and finally displaying them. The Combinatorica user community ranges from students to engineers to researchers in mathematics, computer science, physics, economics, and the humanities. The use of Mathematica in Graph Theory, which is also part of Discrete Mathematics, has been extensively explained by Steven Skiena [15]. It has been perhaps the most widely used software for teaching and research in discrete mathematics since its initial release in 1990. The new Combinatorica is a substantial rewrite of the original 1990 version. It is now much faster than before, and provides improved graphics and significant additional functionality. Combinatorica is included with every copy of Mathematica as `DiscreteMath`Combinatorica`` [26].

4.2   The Concept of web*Mathematica*

One of the most exciting new technologies for dynamic mathematics on the World Wide Web is a web*Mathematica*. It makes all numerics, symbolics, and graphics computing available over the web. It provides an alternative interface via the web. Even in a web environment, the front end is extremely useful. This new technology developed by Wolfram Research that allows the generation of dynamic web content with Mathematica. People use the existing Internet browsers such as Internet Explorer or Netscape as an interface to web*Mathematica* and they do not need to know Mathematica to use it.

There are various important features that Mathematica can offer to a web site, including computation, an interactive programming language, connectivity, the Mathematica frond end, and enhanced support for MathML.

web*Mathematica* is based on a standard Java technology called servlets. Servlets are special Java programs that run on a web server machine. Typically they run in a separate program called a *servlet container*, which connects to the web server. Two popular servlet containers are Tomcat and JRun. Both of them include stand-alone web serves, as well, so they can be used as total solutions themselves without requiring an external web serves such as Apache. web*Mathematica* allows a site to deliver HTML pages that are enhanced by the addition of Mathematica commands. When a request is made for one of these pages the Mathematica commands are evaluated and the computed result is placed in the page. This is done with the standard Java templating mechanism, Java server pages (JSPs) making use of a library of tag extensions called the JSP Taglib. The aim of web*Mathematica* and JSP technology is reduce the amount of extra knowledge required for developing a side to a minimum. The JSP scripts require some knowledge of HTML, including FORM and INPUT elements, and Mathematica but not Java nor JavaScripts [21, 25].

4.3   Edge Coloring for Common Graph Families with web*Mathematica*

We use some commands in the Combinatorica package with Mathematica to color the graphs and to give web-based examples with webMathematica as follows:

```
<%@ page language="java" %>
<%@ taglib uri="/webMathematica-taglib" prefix="msp" %>
<html> <body bgcolor="# ffffff">
<msp:allocateKernel>
<msp:evaluate> <<DiscreteMath`Combinatorica`
</msp:evaluate>
<FORM ACTION="edge.jsp" METHOD="POST">
<INPUT type="text" name="v" ALIGN="LEFT" size="6"
Value="<msp:evaluate> MSPValue[$$v, "3"]</msp:evaluate>" />
Enter the number of vertices of the cycle graph:
<msp:evaluate>
input=True;
If[MSPValueQ[$$v],
n=MSPToExpression[$$v];
c= EdgeColoring[ g = Cycle[n] ];
e=Edges[g];
z=Max[c];
kk = Table[i, {i, 1, 1000}]; For[i = 1, i <= n ,
For[j = 1, j <= M[g], If[c[[j]] = i, kk[[j]] = Hue[i/ z]];
j++]; i++];,input=False;] </msp:evaluate>
</form>
<msp:evaluate> input=True;
If[MSPValueQ[$$v],
MSPShow[ShowGraph[g,Table[e[[i]], EdgeColor->kk[[i]],
i, 1, M[g]], VertexNumber-> On, EdgeStyle-> Thick]],
input=False;]</msp:evaluate>
<INPUT TYPE="Hidden" NAME="formNo" VALUE="1">
<INPUT TYPE="Submit"NAME="taskValue"
VALUE="Color the cycle graph"> </msp:allocateKernel>
</body></html>
```
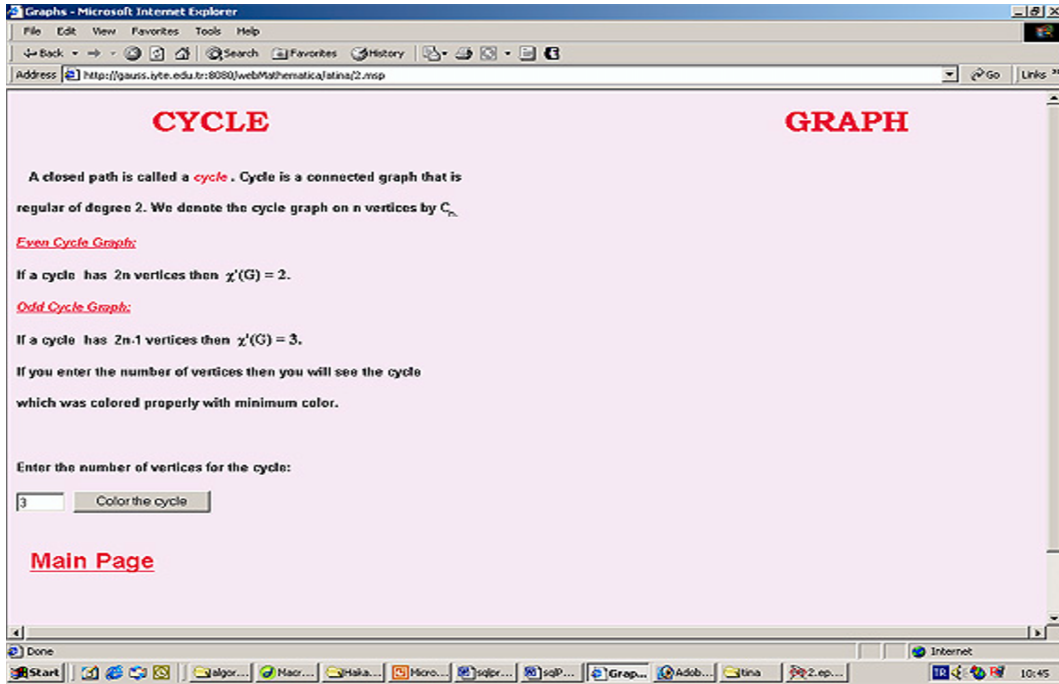
Figure 4.1: A view of the web page before running the program

In this example, there are two <INPUT> tags: one of them allow the user of the page to enter the number of the vertex in the cycle graph, and the second specifies a button that, when pressed, will submit the FORM. When the FORM is submitted, it will send information from INPUT elements to the URL specified by the ACTION attribute; in this case, the URL is the same JSP. Information entered by the user is sent to a Mathematica session and assigned to a Mathematica symbol. Additionally, the Mathlets refer to Mathematica functions that are not in standard usage. In this example some Mathematica commands; If, Table, Edges, M, EdgeColoring, Cycle, ShowGraph, True, False, and some mathematical operations are used by the Mathlets. The name of the symbol is given by prepending $$ to the value of the NAME attribute. MSPValue returns the value of variable or a default if no value exists. This example also demonstrates the use of page scoped variables with MSPToExpression. MSPToExpression interprets values and returns the result. MSPShow saves an image on the server and returns the necessary HTML to refer to this image (see Figure 4.1 and Figure 4.2). The image uses a GIF format; it is possible to save images in other formats [25].
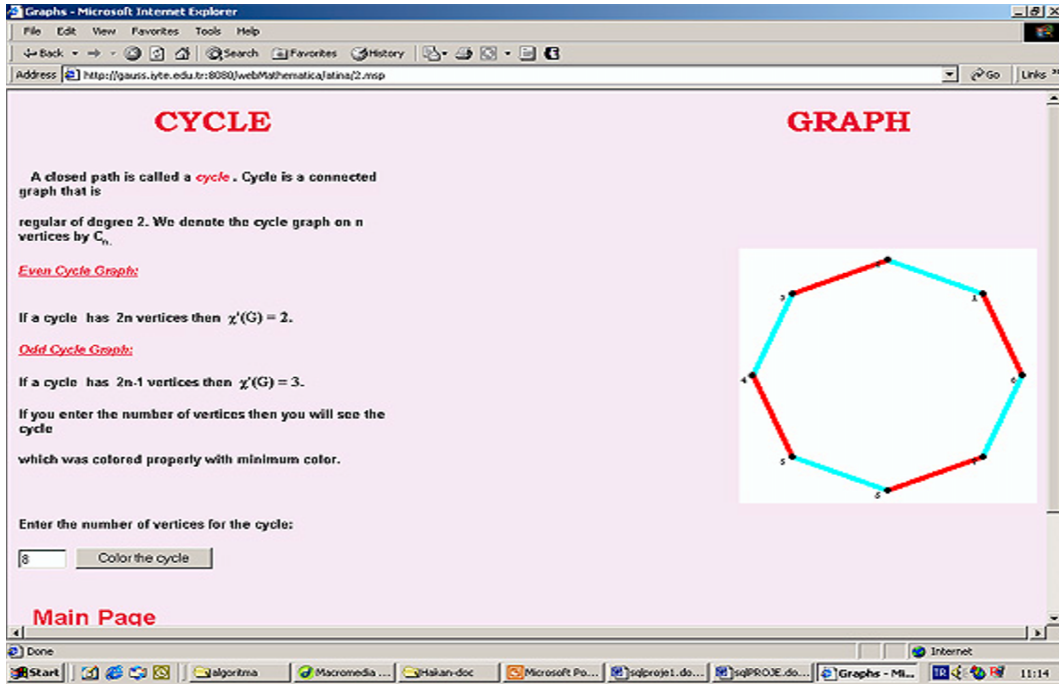
30

Figure 4.2: A view of 2-edge colored cycle graph

If we change the Cycle[n] by Wheel[n], CompleteGraph[n], Star[n] and RandomTree[n], we also color these graphs. For coloring the edges of complete bipartite graph (CompleteKPartiteGraph[n,m]), we need one more INPUT tag.

4.4 Edge Coloring for any graphs with web*Mathematica*

The module DrawG draw the simple graph without isolated points. DrawG takes as input the list of edges of a graph. The vertices of the graph of order n must be labelled consecutively $1, 2, \cdots, n$. This module must be added to the package `DiscreteMath\`Combinatorica\``. The module is;

```
DrawG[elist_]:=Module[{edgelist=elist,size,vertlist,vnum},
size=Length[edgelist];
vertlist=Union[Flatten[edgelist]];
vnum=Length[vertlist];
Do[edgelist[[i]]={edgelist[[i]]},{i,size}];
vertlist=CompleteGraph[vnum][[2]];
Graph[edgelist,vertlist]]
```

31

The following example draws the given graph and colors edges. Let the vertices of a graph be $1, 2, 3, 4, 5$ and the set of edges be $e = \{\{1,2\}, \{2,3\}, \{2,5\}, \{3,4\}, \{3,5\}\}$. Then the graph can be drawn by typing the command ShowGraph[DrawG[e]].

```
<FORM ACTION="bes.msp" METHOD="POST">
<%INPUT type="text" name="v" ALIGN="LEFT" size="50"
Value ="<%Mathlet MSPValue[ $$v, "{{1,2},{2,3},{3,4}}"] %>
<%Mathlet <<DiscreteMath`Combinatorica`%>
<%Mathlet <<Graphics`Colors`%>
<%Mathlet input=True;
If[MSPValueQ[$$v],
s=MSPToExpression[$$v];
c= EdgeColoring[ g = DrawG[s] ];
e=Edges[g];
m=Max[c];
kk =Table[i, {i, 1, 1000}]; For[i = 1, i <= m5 , For[j = 1, j <= M[g5],
If[c5[[j]] = i, kk5[[j]] = Hue[i/m5]]; j++]; i++];, input=False;] %>
<%Mathlet input=True;
If[MSPValueQ[$$v],
MSPShow[ShowGraph[g,Table[e[[i]],
EdgeColor->kk[[i]], i, 1, M[g]],
VertexNumber-> On,EdgeStyle-> Thick]],input=False;] %>
<INPUT TYPE="Hidden" NAME="formNo" VALUE="1">
<INPUT TYPE="Submit"NAME="taskValue"
VALUE="Color the edges of the graph" > </FORM>
```

A form element is a block of HTML that may contain input elements. A form may be activated with an input of type submit. The action attribute refers to an URL that accessed when the form is activated. The method attribute tells the browser what HTTP method to use, in this case, a post method. This example has two input tags. The first allows the user of the page to enter the list of edges of the graph, and the second specifies a button that, when pressed, will submit the form. When the form is submitted, it will send information from input elements to the URL specified by the action attribute. This information is sent to a Mathematica kernel and assigned to a Mathematica symbol (see Figure 4.3). The name of the symbol is given by $$ to the value of the name attribute. When a value entered in the text field and the <Color the graph's edges and

vertices> button pressed, the text is displayed. This example also shows the use of the MSP functions MSPShow, MSPValue, and MSPToExpression. [20]
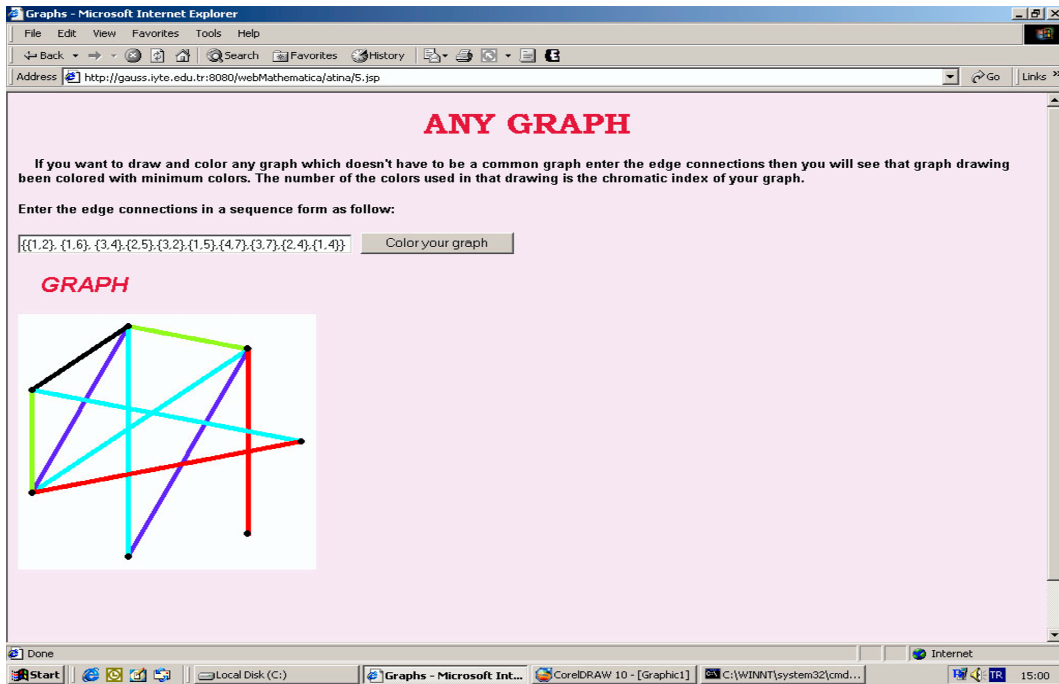

Figure 4.3: A view of edge-colored for any graph

To color edges of the standard graphs (complete graph, wheel, tree, cycle, and the others) we need more inputs. If the web user selects one of the standard graphs and its number of vertices then they can get easily colored graph and its chromatic index as follows:

```
<FORM ACTION="familycolor.jsp" METHOD="POST">
<p>Please select one of the following graphs and
input into the box:(1,2,3,or 4) </p> <p>1-CompleteGraph,</p>
<p>2-RandomTree, </p> <p>3-Wheel,</p> <p>4-Cycle)</p>
<msp:allocateKernel>
<INPUT type="text" name="m" ALIGN="LEFT" size="6"
value="<msp:evaluate> MSPValue[$$m,"1"]</msp:evaluate>" />
Input the number of the vertices for the selected graph:
<INPUT type="text" name="n" ALIGN="LEFT" size="6"
value="<msp:evaluate> MSPValue[$$n,"5"]</msp:evaluate>" />
<msp:evaluate> <<DiscreteMath`Combinatorica`
<<Graphics`Colors`</msp:evaluate>
<msp:evaluate> MSPBlock[{$$m,$$n},
Which[$$m==1, MSPShow[ColorEdges[CompleteGraph[$$n]]],
$$m==2, MSPShow[ColorEdges[RandomTree[$$n]]],
$$m==3, MSPShow[ColorEdges[Wheel[$$n]]],
$$m==4, MSPShow[ColorEdges[Cycle[$$n]]]]]
</msp:evaluate>
```

33

```
The Chromatic Index of the selected graph is:
<msp:evaluate>
MSPBlock[{$$m,$$n},
Which[$$m==1, EdgeChromaticNumber[CompleteGraph[$$n]],
$$m==2, EdgeChromaticNumber[RandomTree[$$n]],
$$m==3, EdgeChromaticNumber[Wheel[$$n]],
$$m==4, EdgeChromaticNumber[Cycle[$$n]]]]
</msp:evaluate> </p> <input type="submit" name="button"
value="Graph Color"> </msp:allocateKernel>
```

Combinatorica package uses Brelaz's heuristic to find a good, but not necessarily minimal for edge coloring of graph G. Then when the web user enters the huge number of the vertices he / she might get interesting results or time out error for the edge coloring.

REFERENCES

[1] S. D. Andres, The Game Chromatic Index of Forests of Maximum Degree 5, *Electronic Notes in Discrete Mathematics*, (2003)

[2] R.Balakrishnan, K.Ranganathan, A Textbook of Graph Theory, Springer-Verlag, (2000)

[3] L. W. Beineke, R. J. Wilson , On the edge-chromatic number of a graph, *Discrete Mathematics*, 5, (1973), P: 15-20

[4] H. L. Bodlaender, On the complexity of some coloring games, *Lecture Notes in Computer Science*, 484, (1991), P: 30-40

[5] L. Cai, X. Zhu, Game Chromatic Index of k-degenerate Graphs, *J. Graph Theory*, 36, (2001), P: 144-155

[6] G. Chartrand, L. Lesniak, Graphs & Digraphs, Chapman & Hall / CRC, (2000)

[7] P. L. Erdös, U. Faigle, W. Hochstättler and W. Kern, Note on the Game Chromatic Index of Trees, *Theoretical Computer Science*, (2003)

[8] S. Fiorini, R. J. Wilson, Edge-Colouring of Graphs, Pitman, (1977)

[9] J. Gross, J. Yellen, Graph Theory and Its Applications, CRC Press, (1999)

[10] A. J. Hilton, Recent Progress in Edge-Colouring Graphs, *Discrete Mathematics*, 64, (1987), P: 303-307

[11] I. J. Holyer, The NP-Completeness of Edge Colouring, *SIAM Journal on Computing*, 10, (1981), P: 718-720

[12] D. König, Über Graphen und iher Anwendung auf Determinantentheorie und Mengenlehre, *Math. Ann.*, 77, (1916), P: 453-465

[13] P. C. B. Lam, W. C. Shiu and B. Xu, Edge Game Coloring of Graphs, *Graph Theory Notes of New York*, 37, (1999), P: 17-19

[14] S. Nakano, T. Nishizeki, Edge-Coloring Problems for Graphs, *Interdisciplinary Information Sciences*, 1, (1964), P: 19-32

[15] S. Pemmaraju, S. Skiena, Computational Discrete Mathematics-Combinatorics and Graph Theory with Mathematica, Cambridge University Press, (2003)

[16] L. Perkovic, Edge Coloring, Polyhedra and Probability, *Ph.D. Thesis in Carnegie-Mellon University* , (1998)

[17] R. Rizzi, Finding 1-Factors in Bipartite Regular Graphs and Edge-Coloring Bipartite Graphs, *SIAM Journal on Discrete Mathematics*, 15, (2002), P: 283-288

[18] C. E. Shannon, A theorem on coloring the lines of a network, *J. Math. Phys.*, 28, (1949), P: 148-151

[19] K. Thulasiraman, M. N. S. Swamy, Graphs:Theory and Algorithms, John Wiley & Sons, Inc., (1992)

[20] Ü.Ufuktepe, G.Bacak, T. Beseri, Graph Coloring with web*Mathematica*, *Lecture Notes in Computer Science* , 3039, (2004), P: 376-381

[21] Ü.Ufuktepe, An Application with web*Mathematica*, *Lecture Notes in Computer Science* , 2657, (2003), P: 774-780

[22] V. G. Vizing, On an estimate of the chromatic class of a p-graph, *Discret. Analiz*, 3, (1964), P: 25-30

[23] V. G. Vizing, The chromatic class of a multigraph, *Kibernetica (Kief)*, 3, (1965), P: 167-175

[24] V. G. Vizing, Critical graphs with given chromatic class, *Metody Diskret. Analiz*, 5, (1965), P: 9-17.

[25] T. Wickham, webMathematica A User Guide, Wolfram Research Inc., (2002)

[26] S. Wolfram, The Mathematica Book, Cambridge Univ. Press, (1996)