The Structure of Bipartite Graphs

RESEARCH PROJECT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF DEGREE OF BACHELOR OF EDUCATION IN MATHEMATICS

By:

Hawraa Kareem Wali

Supervisor:

Dr. Murtadha Ali

DEPARTMENT OF MATHEMATICS

COLLEGE OF EDUCATION

UNIVERSITY OF MISSAN

2024

بِسْمِ اللَّـهِ الرَّحْمَـٰنِ الرَّحِيمِ

(كهيعص)

﴿إِنَّا فَتَحْنَا لَكَ فَتْحًا مُبِينًا (١) لِيَغْفِرَ لَكَ اللَّهُ مَا تَقَدَّمَ مِنْ ذَنْبِكَ وَمَا تَأَخَّرَ وَيُتِمَّ نِعْمَتَهُ عَلَيْكَ وَيَهْدِيَكَ صِرَاطًا مُسْتَقِيمًا (٢) وَيَنْصُرَكَ اللَّهُ نَصْرًا عَزِيزًا (٣)﴾

لطالما رافقتني هذه ألايات في
مراحل حياتي الدراسيه وأنا أريد أن
أبدأ وأختم بهما هذه المسيره
الدراسيه

أهداء

الى من القلوب أحترقت شوقا لرؤيته الى من الأرض بأنتظاره ليملأها قسطاوعدلا بعد ما ملئت ظلماوجورا الى مصباح الهدى وسفينه النجاه الى الذي كانت كلمه ألهي بالحسين الشهيد تسبق كل خطواتي الى من أضاءوا بعلومهم عقول غيرهم وأهدو بالجواب حيره السائلين وعصمه من تمسك بهم ونجاه من أتبعهم الى الأثني عشر نورا (عليهم السلام)

الى الركن الدافئ في قلبي الى من كانت دعواتها سر توفيقي...(أمي)

الى من علمني أن الحياه تدوس الضعفاء فكنت بكلماته أعود الى قوتي... (أبي)

الى رفاق الخطوه الأولى والخطوه ماقبل الأخيره الى من كانوا في السنوات العجاف سحابا ممطرا

شكر وتقدير ...

A list Of Symbols

G	Graph
V	Set of vertices
E	Set of edges
$Deg(v)$	Degree of vertex v

Abstract

Bipartite graphs are a fundamental concept in graph theory, representing relationships between two distinct sets of vertices with no edges within the same set. In this project, we delve into the theory of bipartite graphs, exploring their properties, characteristics, and practical applications.

We begin by providing a comprehensive overview of graphs, covering basic definitions, terminologies, and structural properties. Through theoretical analysis and illustrative examples, we discuss key concepts such as bipartite matching, bipartite connectivity, and the Hall's theorem.

Next, we examine the properties and characteristics of bipartite graphs, including their maximum matching size, minimum vertex cover, and chromatic number. We explore various algorithms for solving bipartite graph problems, such as the Hungarian algorithm for maximum bipartite matching.

Contents

Chapter 1

Introduction

1.1 Historical Background:

Graph theory is a branch of mathematics that deals with the study of graphs, which are mathematical structures used to model pairwise relations between objects. The history of graph theory spans several centuries and has evolved through contributions from various mathematicians and researchers. Here's a brief historical background:

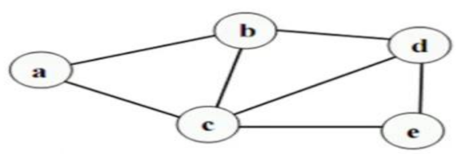
- 18th Century: Leonhard Euler introduced graph theory while solving the Seven Bridges of Königsberg problem.
- 19th Century: Gustav Kirchhoff and Arthur Cayley made foundational contributions.
- 20th Century: Frank Harary, Paul Erdős, and others expanded graph theory, tackling problems like graph coloring and planarity.
- 21st Century: Graph theory found applications in diverse fields like computer science and biology, with a focus on complex networks.

1.2 Fundamental Definitions and Notations:

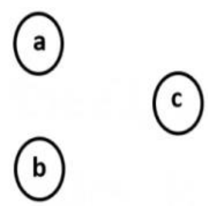
Graphs 1.2.1: graphs are discrete structures consisting of vertices and edges that connect these vertices. A graph $G = (V, E)$ consists of V, a nonempty set of vertices (or nodes) and E, a set of edges. Each edge has either one or two vertices associated with it, called its endpoints. An edge is said connect its endpoints. A graph is visualized by using points to represent vertices and line segments to represent edges. A graph with an infinite vertex set or an infinite number of edges is called an infinite graph and the graph with finite vertex set and finite number of edges is called finite graph. Again a graph denoted as $G = (V, E)$ consists of a non-empty set of vertices or nodes V and a set of edges E

- $V(G)$: a finite , non-empty set of vertices.
- $V(G)$: a set of edges (pairs of vertices).
- An edge is a path between two nodes.

Example 1.2.2: $V = \{a, b, c, d, e\}$, $E = \{(c, a), (c, b), (c, d), (c, e), (a, b), (b, d), (d, e)\}$



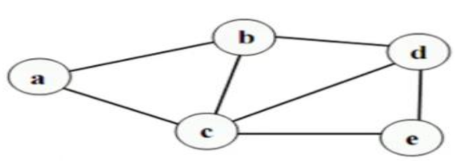
Null Graph 1.2.3: A graph having no edges is called null graph.



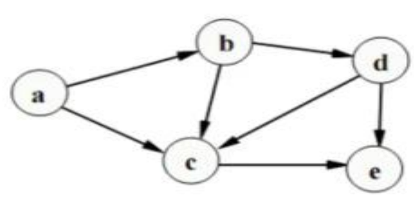
Trivial Graph 1.2.4: A graph with only one vertex is called a trivial graph.

Undirected and Directed Graphs 1.2.5: When the edges in a graph have no direction then the graph is called undirected, if the edges in a graph have direction then the graph is called directed.

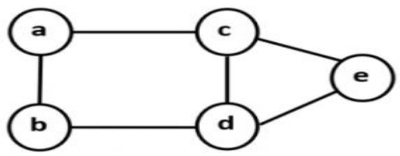
Example 1.2.6: $V = \{a, b, c, d, e\}$, $E = \{(c, a), (c, b), (c, d), (c, e), (a, b), (b, d), (d, e)\}$



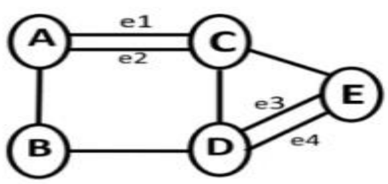
Example 1.2.7: $V = \{a, b, c, d, e\}$, $E = \{(a, b), (a, c), (b, c), (b, d), (d, c), (d, e), (c, e)\}$



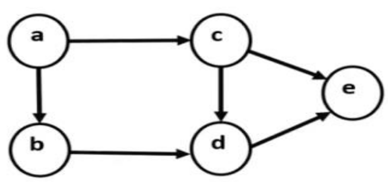
Simple Graph 1.2.8: A graph in which each edge connects two different vertices and where no two edges connect the same pair of vertices is called a simple graph.



Multi-Graph 1.2.9: A graph containing at least one pair of vertices $\{u, v\}$ that have multiple edges is called a multi-graph.

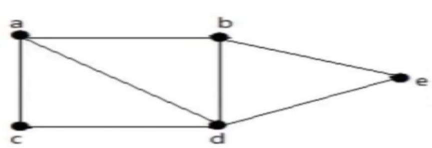


Simple Directed Graph 1.2.10: A directed edge graph that has no loops and has no multiple directed edges is called a simple directed graph.



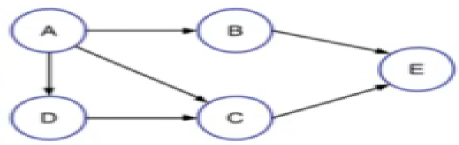
Connected and Disconnected Graph 1.2.11: A graph is said to be connected if there is a path between every pair of vertices. From every vertex to any other vertex, there should be some path to traverse.

Example 1.2.12: In this example, it is possible to travel from one vertex to any other vertex. For example, one can traverse from vertex a to vertex e using the path $a - b - e$.



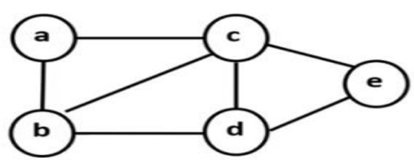
Example 1.2.13: In this example, traversing from vertex a to vertex f is not possible because there is no path between them directly or indirectly. Hence it is a disconnected graph.

Example 1.2.14: This graph is disconnected because from vertex E we cannot visit any other vertex, for example we cannot traverse from vertex E to B or C or any other vertex, so it is disconnected.

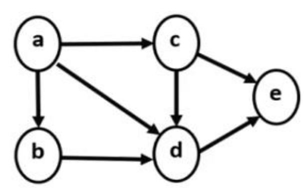


Degree in Undirected Graph 1.2.15 : The degree of vertex v, is defined as the number of adjacent (incident) edges to vertex v in G.

Example 1.2.16: In this graph, $Deg\ (a) = 2$, $Deg\ (b) = 3$.



Degree in Directed Graph 1.2.17 : In directed graph, we have to find two types of degree, in degree and out degree. $In - Deg(v)$ is the number of incoming edges to vertex v. $Out - Deg(v)$ is the number of outgoing edges to vertex v.

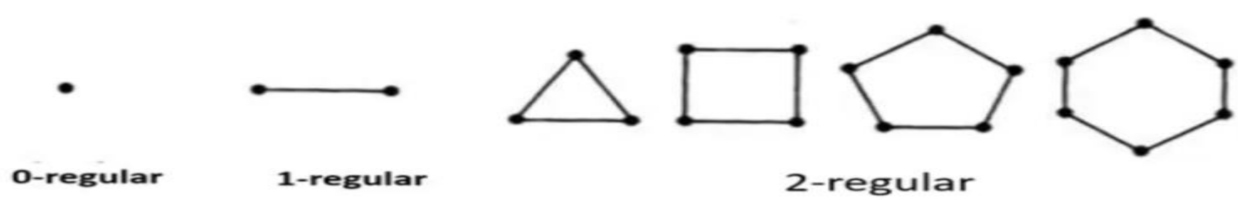


Example 1.2.18: In this graph:

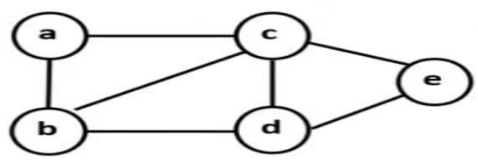
$In - Deg\ (c) = 1$, $In - Deg\ (d) = 3$.

$Out - Deg\ (c) = 2$, $Out - Deg\ (d) = 1$.

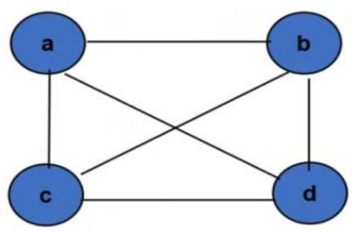
Regular Graph 1.2.19: Regular graph is a graph where each vertex has the same number of neighbors (every vertex has the same degree).



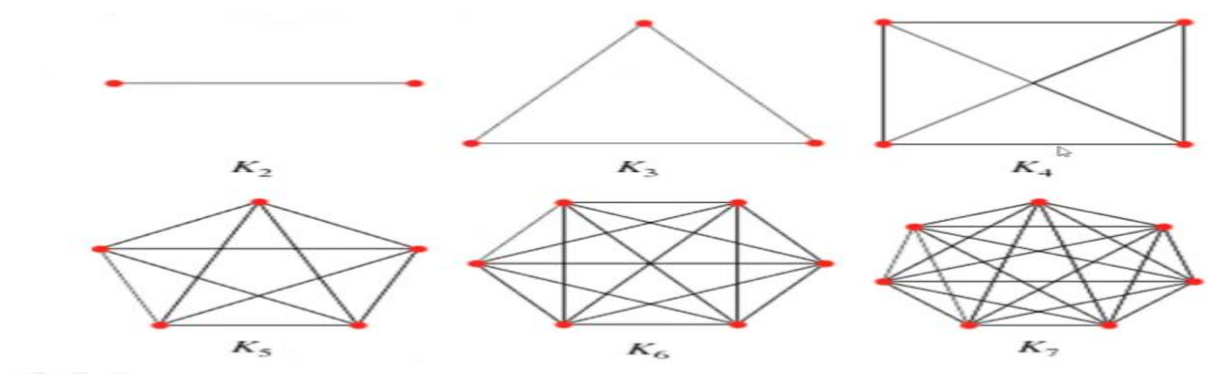
Adjacency 1.2.20: In a graph, two vertices are said to be adjacent, if there is an edge between the two vertices. Here, the adjacency of vertices is maintained by the single edge that is connecting those two vertices. In a graph, two edges are said to be adjacent if there is a common vertex between the two edges. Here the adjacency of edges is maintained by the single vertex that is connecting two edges.



Complete Graph 1.2.21: A graph G is complete graph if and only if every vertex a in G is adjacent to every other vertex b in G. A complete graph is already connected. In general, the number of edges in complete graph is $n(n-1)/2$, where n is the number of vertices.

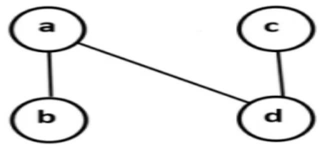


Example 1.2.22: Here is different types of complete graph.

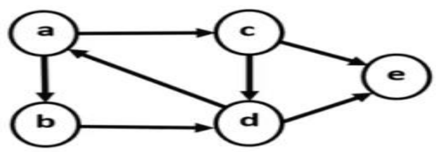


Cyclic Graph 1.2.23: A graph is said to have a cycle if we start from a vertex and after traversing some vertices, we come to the same vertex, then we can say that the graph is having a cycle. If there is a cycle in a graph, then that graph is called cyclic graph. If there is no cycle present in the graph, then the graph is called Acyclic graph. For a cyclic graph, at least one cycle is necessary.

Example 1.2.24: Here there is no cycle, so the graph is Acyclic

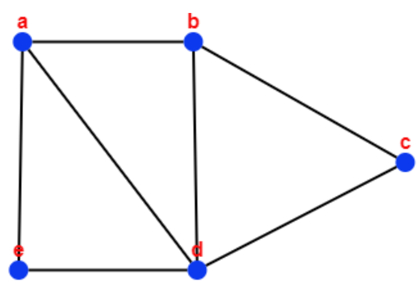


Example 1.2.25: In this graph, we have more than one cycle, so the graph is cyclic.



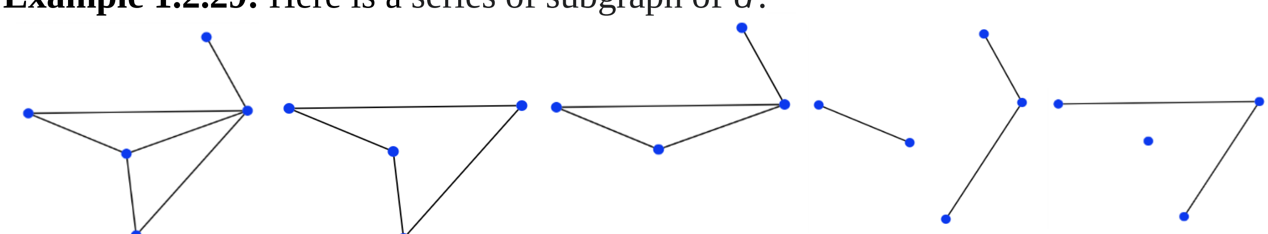
Path of Graph 1.2.26: A **walk** of length k in a graph is a succession of k edges joining two vertices. A **trail** is walk in which all the edges (but not necessarily all the vertices) are distinct. A **path** is a walk in which all the edges and all the vertices are distinct.

Example 1.2.27: In this graph, $abdcbde$ is a walk of length 6 between a and e. It is not a trail (because edge bd is traversed twice). The walk $adcbde$ is a trail length 5 between a and e. It is not a path (because vertex d is visited twice). The walk $abcde$ is a path of length 4 between a and e.



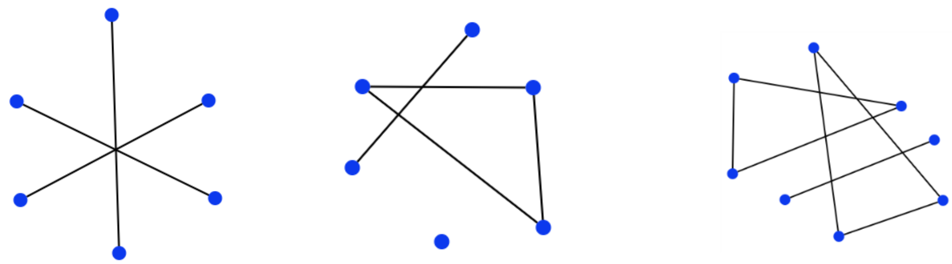
Subgraph of Graph 1.2.28: A subgraph of G is a graph all of whose vertices and edges are vertices and edges of G.

Example 1.2.29: Here is a series of subgraph of G.



Components 1.2.30: Every disconnected graph can be split into a number of connected subgraphs called its components. It may not be immediately obvious that a graph is disconnected.

Example 1.2.31: This figure 4 shows 3 graphs, each disconnected and comprising 3 components.



Chapter 2

Bipartite Graph and Complete Bipartite Graph

2.1. Bipartite graph

A bipartite graph, also known as a bigraph, is a special type of graph that consists of two disjoint sets of vertices, where edges only connect vertices from different sets. In other words, the vertices of a bipartite graph can be divided into two distinct groups such that all edges connect vertices from one group to the other. This property makes bipartite graphs particularly useful in modeling and solving various real-world problems.

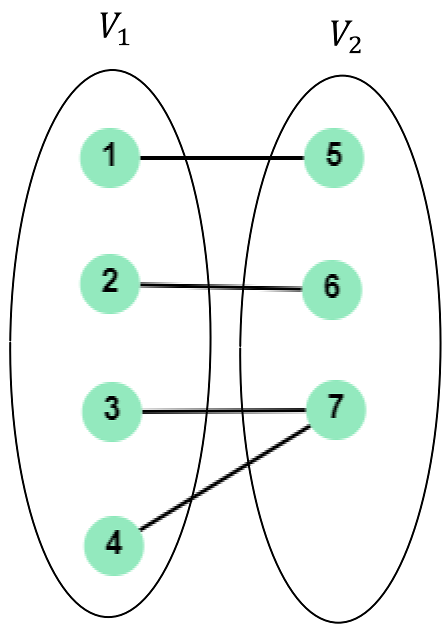
Bipartite graphs have a wide range of applications in diverse fields such as computer science, mathematics, social sciences, and biology. They are commonly used to represent relationships or interactions between two distinct types of entities. For example, in social network analysis, bipartite graphs can be used to model relationships between users and various entities such as products, interests, or organizations.

Let $G = (V, E)$ is a graph then G is a bipartite if and only if there exists a partition of its vertex set into two disjoint sets will call V_1 and V_2 such that every edge in the graph joins a vertex from one set to a vertex in the other set, this means that there are no adjacent vertices in V_1 and there no adjacent vertices is in V_2. If two vertices are adjacent then one of the vertices is in V_1 and the other is in V_2. If we can find a partition like this :

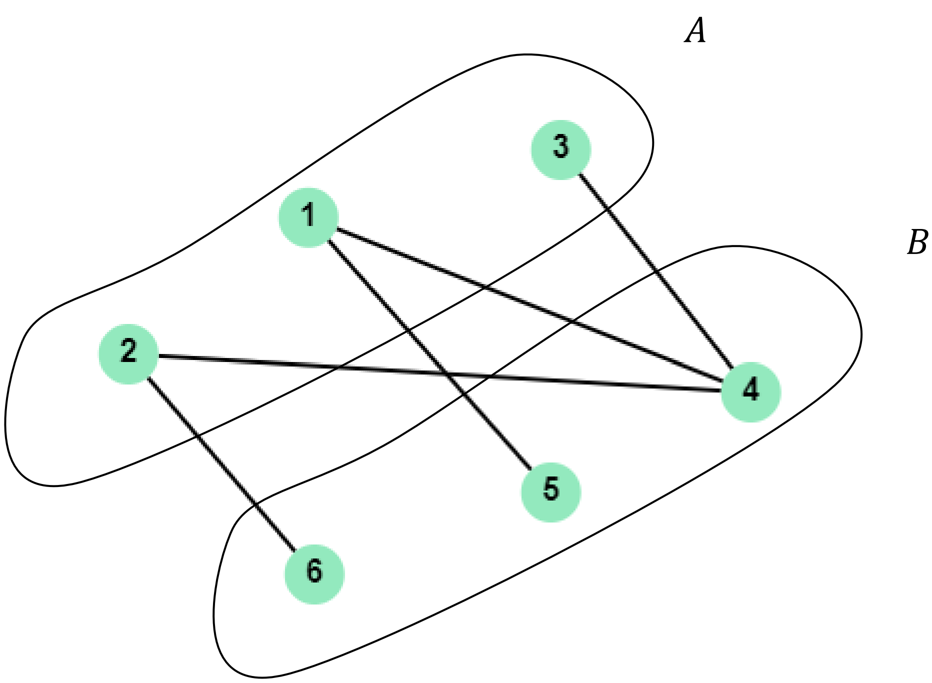
$$V = V_1 \cup V_2$$

of a graph vertices then it is a bipartite graph . Again, a graph is bipartite if we can separate its vertex set or partition its vertex set these two sets we call V_1 and V_2.

Example 2.1.1: This graph is bipartite because we can partition its vertices in this way, so that every edge in the graph goes from V_1 to V_2, every edge joins a vertex from V_1 to a vertex in V_2.



Example 2.1.2: In this graph, we partition its vertices in this way then indeed that its bipartite graph because every edge goes from a set A to the a set B.



2.2. Complete bipartite graph

A complete bipartite graph is a special type of bipartite graph where every vertex in one set is connected to every vertex in the other set. In other words, if we have two disjoint sets of vertices, often denoted as V_1 and V_2, a complete bipartite graph contains an edge between every vertex in V_1 and every vertex in V_2, with no additional edges between vertices within the same set. Complete bipartite graphs are often represented as $K_{m,n}$, where m and n represent the number of vertices in the two sets, V_1 and V_2, respectively. The graph contains m vertices in V_1 and n vertices in V_2, resulting in a total of $m\ ,n$ edges.

Complete bipartite graphs have several interesting properties and applications. One important property is that they have the maximum possible number of edges for a given number of vertices in each set. This property makes complete bipartite graphs useful in various combinatorial problems, such as network flow, matching, and optimization.

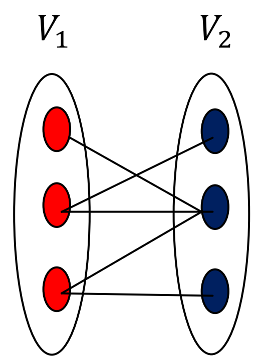
Complete bipartite graphs find applications in diverse fields such as computer science, operations research, and social sciences. For example, in computer science, complete bipartite graphs are used to model and solve problems like assignment and scheduling. In social sciences, they can represent relationships between two distinct groups of entities, such as students and courses, or employees and projects.

Let's say we have got a graph G such that $G = (V, E)$. We say that G is a bipartite graph if there exists a partitioning of its vertex set into two sets we called V_1 and V_2 such that every edge of G joins a vertex in V_1 to a vertex in V_2, or equivalently no edge of the graph joins two vertices that are in the same set. Thus :

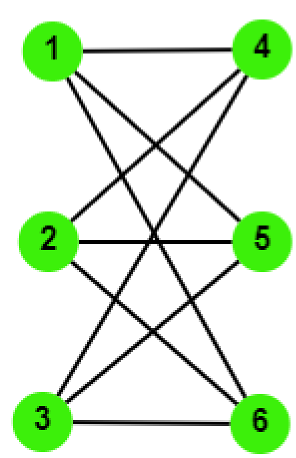
$$V = V_1 \cup V_2$$

Again, this is a partitioning of the vertex set, so the vertex set V equal to the part set V_1 union with the parti set V_2 to say that the partition of V of the vertex set means that every element of the vertex set is in exactly one of these two part sets V_1 and V_2, so they have no common elements.

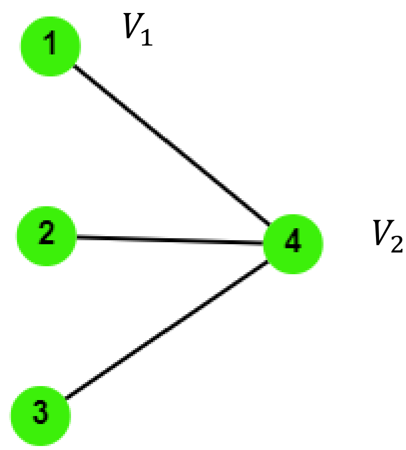
Example 2.2.1: In this example we got 6 vertices in two different colors. Now, we will draw in some black edges joining some of these red vertices to some of the blue vertices. So this graph is bipartite because we see every edge of the graph from part set V_1 goes to the other V_2. Every edge joins a vertex from one part set to a vertex in the other part, so each of these partitioning sets is called independent vertex set because it's a set of vertices where no two vertices it contains are adjacent. Now, is this a complete bipartite graph? This is not a complete bipartite graph because it doesn't have every possible edge. In a complete bipartite graph not only does every edge join a vertex from one part set to the other partitioning set but additionally any pair of vertices in different part sets must be joined by an edge in a complete bipartite graph, so it's very similar to the idea of normal complete graph, it's just a bipartite graph with every possible edge.



Example: Consider the following graph, we can notice that each vertex in this graph is adjacent to 3 vertices in the other part set.

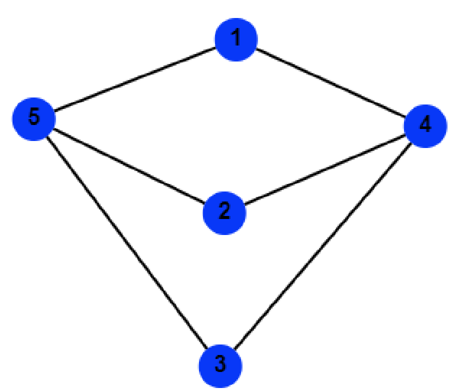


If we deleted two vertices from one part set, we notice all of vertices in one part set are adjacent to the one vertex in the other part set, so they all have degree one but the vertex in the other part set is adjacent to all 3 vertices in the first part set V_1 and so the vertex is has degree 3 in V_2. Then in general of course in a complete bipartite graph the degree of vertex in a part V_2 set will be equal to the cardinality of the other part set V_1.



Notation: The graph K_n denotes a complete graph on n −vertices, so K_1 tells us it's a complete graph and $K_{3,1}$ tells us that it is a complete bipartite graph where one part set will have 3 vertices and the other part set will have one vertex, while $K_{1,3}$ tells us that it is a complete bipartite graph where one part set will have 1 vertex and the other part set will have 3 vertices.

Example 2.2.2: In this example will draw $K_{3,2}$ graph, this graph is a complete bipartite graph with part sets of cardinality 3 and 2. Each vertex in part set V_2 has degree 3, they are adjacent to the 3 vertices in the other part set, whereas each vertex in part V_1 has degree 2, they are each adjacent to the 2 other vertices in the other part set, so that what a complete bipartite graph is it. It is a bipartite graph with every possible edge, so every pair of vertices that are in different part sets must be joined by an edge that is a complete bipartite graph.



Chapter 3

Results and Discussions

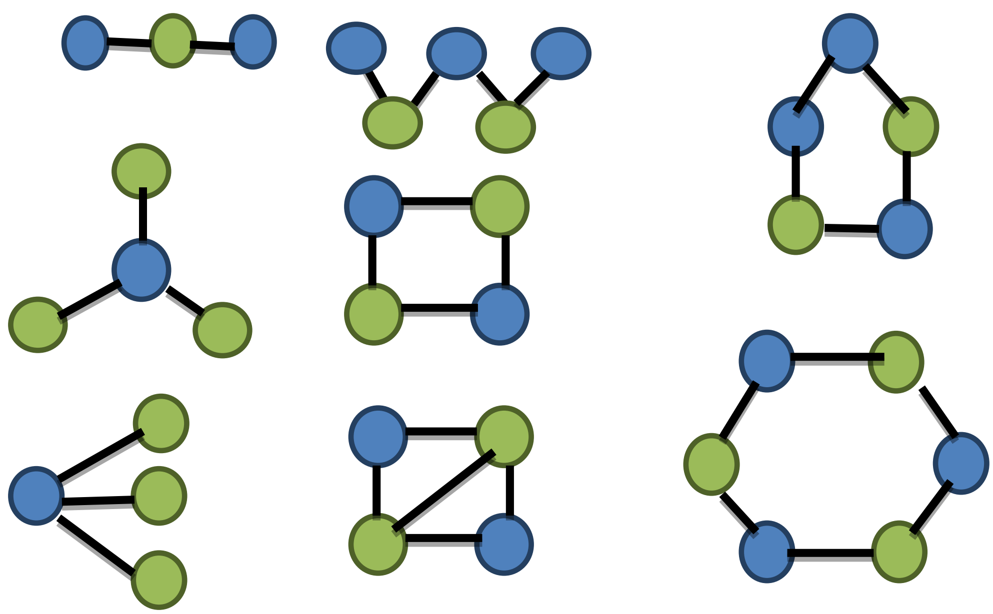
- **Bipartite Graph Algorithm.**

The following algorithm determines whether a given graph is bipartite or not.

Input: The graph G(V, E) and a starting vertex S as input.

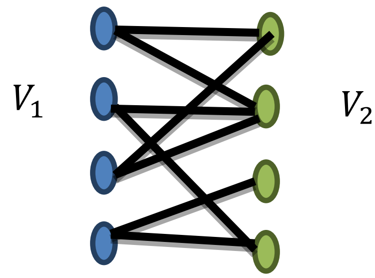
The algorithm returns either the input graph G is bipartite or the graph is not a bipartite graph.

1. Assign a blue color to the starting vertex S.
2. Find the neighbors of the starting vertex and assign a green color.
3. Find the neighbor's neighbor and assign a blue color.
4. Continue this process until all the vertices in the graph are assigned a color.
5. During this process, if a neighbor vertex and the current vertex have the same color then the algorithm terminates. The algorithm returns that the graph is not a bipartite graph.



- **If $\mathbf{G(V, E)}$ is a bipartite graph then the sum of degrees of vertices of each set is equal to the number of edges.**

$$\sum_{a \in V_1} \boldsymbol{deg(a)} = \sum_{a \in V_2} \boldsymbol{deg(a)} = |\boldsymbol{E}|$$



$\sum_{a \in V_1} deg(a) = \sum_{a \in V_2} deg(a) = |E| = 8.$

- **Every tree is a bipartite graph.**

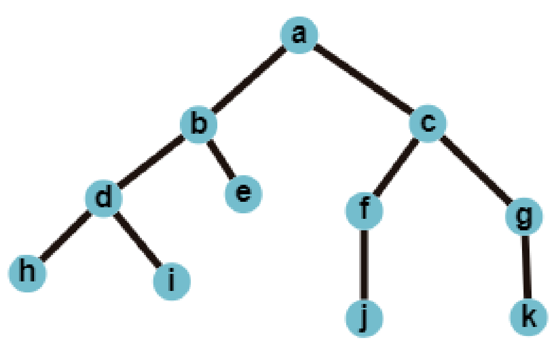
Proof: We want to take a look at two possible algorithms to show any tree is bipartite.

To show the graph is a bipartite we must divide the vertices into two sets A and B, so that no two vertices in the same set are adjacent. Here is the first algorithm we can use to show any tree is bipartite.

Here is the steps:

Step 1: Designate any vertex as the root, put this vertex in a set A.

Looking at the following tree:



We will designate vertex $\boldsymbol{a}$ as the root, vertex $\boldsymbol{a}$ goes in a set A. Let's go ahead and cross out the vertices, as we place them in the two sets.

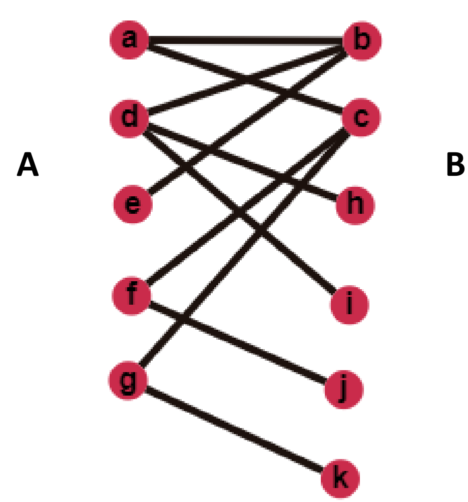
Step 2: Put all the children of the root in a set B, none of these children are adjacent, they are siblings. $\boldsymbol{b}$ and $\boldsymbol{c}$ children of $\boldsymbol{a}$**,** $\boldsymbol{b}$ and $\boldsymbol{c}$ go in a set B. So we have an edge connecting $\boldsymbol{a}$ and $\boldsymbol{b}$**,** and an edge connecting $\boldsymbol{a}$ and $\boldsymbol{c}$**.**

Step 3: Put into a set A every child of every vertex in B (i.e. every grandchild of the root). So looking at vertex $\boldsymbol{b}$, vertices $\boldsymbol{d}$ and $\boldsymbol{e}$ are children of $\boldsymbol{b}$. Since $\boldsymbol{b}$ in a set B, $\boldsymbol{d}$ and $\boldsymbol{e}$ go in a set A. We have an edge connecting $\boldsymbol{b}$ and $\boldsymbol{d}$ as well as $\boldsymbol{b}$ and $\boldsymbol{e}$. And now**,** we go back to vertex $\boldsymbol{c}$, $\boldsymbol{f}$ and $\boldsymbol{g}$ are all children of $\boldsymbol{c}$, because $\boldsymbol{c}$ is in a set B we place $\boldsymbol{f}$ and $\boldsymbol{g}$ in a set A, and we have an edge connecting $\boldsymbol{c}$ and $\boldsymbol{f}$ as well as an edge connecting $\boldsymbol{c}$ and $\boldsymbol{g}$.

Step 4: All vertices have been assigned one of sets, alternating between A and B every generation. Now let's go back over to vertex $\boldsymbol{d}$ which is in a set A. $\boldsymbol{h}$ and $\boldsymbol{i}$ are children of $\boldsymbol{d}$. Because $\boldsymbol{d}$ is in the set A then $\boldsymbol{h}$ and $\boldsymbol{i}$ go in a set B, and we have an edge connecting $\boldsymbol{d}$ and $\boldsymbol{h}$ as well as $\boldsymbol{d}$ and $\boldsymbol{i}$. Now let's go back to vertex $\boldsymbol{f}$, $\boldsymbol{j}$ is a child of $\boldsymbol{f}$ and $\boldsymbol{f}$ is in a set A, therefore $\boldsymbol{j}$ goes in a set B and we have an edge connecting $\boldsymbol{f}$ and $\boldsymbol{j}$.

Finally $\boldsymbol{k}$ is a child of $\boldsymbol{g}$, $\boldsymbol{g}$ is in a set A and therefore $\boldsymbol{k}$ goes in a set B and we have an edge connecting $\boldsymbol{g}$ and $\boldsymbol{k}$. In short, A vertex is in a set B if and only if it is the child of a vertex in a set A.

This shows the given tree is a bipartite graph. ■



- **A graph G is bipartite if and only if it has no odd cycles.**

Proof: **First direction:**

Let $G = (V, E)$ be bipartite.

So, let $V = A \cup B$ such that $A \cap B = \emptyset$ and that all edges $e \in E$ are such that e is of the form $\{a, b\}$ where $a \in A$ and $b \in B$, this is the definition of bipartite graph.

Suppose that G has at least on odd cycle C.

Let the length of C is n.

Let $C = (v_1, v_2, \dots, v_n, v_1)$.

Let $v_1 \in A$. It follows that $v_2 \in B$ and hence $v_3 \in A$, and so on.

Hence we see that $\forall k \in \{1,2, \dots, n\}$ we have:

$$v_k \in \begin{cases} A: k \text{ is odd} \\ B: K \text{ is even} \end{cases}.$$

But as n is odd, $v_n \in A$.

But $v_n \in A$, and $v_n v_1 \in C_n$.

So, $v_n v_1 \in E$ which contradicts the assumption that G is bipartite.

Hence if G is bipartite, it has no odd cycles.

Second direction:

Suppose that G has no odd cycles.

Choose any vertex $v \in G$.

Divide G into two sets of vertices like this:

Let A be the set of vertices such that the shortest path from each element of A to v is of odd length.

Let B be the set of vertices such that the shortest path from each element of B to v is of Even length.

Then $v \in B$ and $A \cap B = \emptyset$.

Suppose that $a_{1,} a_2 \in A$ are adjacent.

Then there would be a walk a closed walk of odd length $(v, \dots, a_1, a_2, \dots, v)$.

But from graph containing closed walk of odd length also contains odd cycle, it follows that G would contain an odd cycle. This contradicts from supposition that G contains no odd cycles. So no two vertices in A can be adjacent.

By the same argument, neither can any two vertices in B be adjacent.

Thus A and B satisfy the conditions for $G = (A \cup B, E)$ to be bipartite. ■

REFERENCES

- Fei, Hong. (2023). Graph Theory. doi: 10.1007/978-3-031-26212-8_7.
- (2023). Graph Theory. Oberwolfach Reports, doi: 10.4171/owr/2022/1.
- (2023). Fundamentals of Graph for Graph Neural Network. Advances in systems analysis, software engineering, and high performance computing book series, doi: 10.4018/978-1-6684-6903-3.ch001.
- Niswah, N., Qonita., Yeni, Susanti. (2023). Bipartite graph associated with elements and cosets of subrings of finite rings. Barekeng, doi: 10.30598/barekengvol17iss2pp0667-0672.