MATH 341- Honors Project Spring 24 (Graph Theory)

EXPOSITORY PAPER: GRAPH THEORY

1) Introduction

Graph theory, a fundamental field within discrete mathematics, studies graphs—mathematical structures used to model pairwise relations between objects [2]. This paper aims to elucidate core concepts of graph theory, including its origins, basic definitions, types of graphs, and its application to solving real-world problems. By providing a clear exposition of these concepts, readers will gain insight into how graph theory underpins many aspects of our daily lives.

Graphs serve as mathematical models to analyse many concrete real-world problems successfully. Certain problems in physics, chemistry, communication science, computer technology, genetics, psychology, sociology, and linguistics can be formulated as problems in graph theory [4].

Also, many branches of mathematics, such as group theory, matrix theory, probability, and topology, have close connections with graph theory. Some puzzles and several problems of a practical nature have been instrumental in the development of various topics in graph theory [7].

2) Historical Context

The history of graph theory may be specifically traced to 1735, when the Swiss mathematician Leonhard Euler solved the Königsberg bridge problem. The Königsberg bridge problem was an old puzzle concerning the possibility of finding a path over every one of seven bridges that span a forked river flowing past an island—but without crossing any bridge twice. Euler argued that no such path exists.

His proof involved only references to the physical arrangement of the bridges, but essentially, he proved the first theorem in graph theory [7].

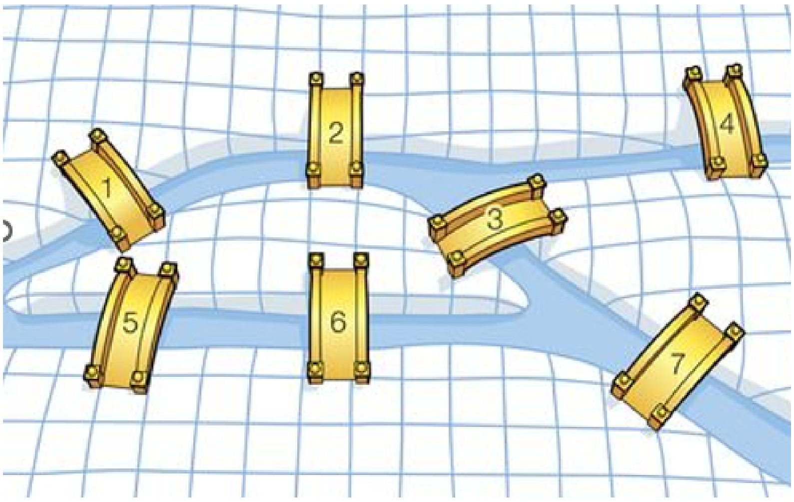

Fig. 1- The 7 bridges of Königsberg (Carlson)

3) Basic Concepts and Definitions

Now we are going to look at some key concepts and their involved definitions in the area of graphs.

a) **Graph (*G*):** A graph G is an ordered pair *G = (V, E)* comprising a set *V* of vertices nodes together with a set *E* of edges or arcs that link pairs of vertices [2].

b) **Vertex (*V*):** A vertex (also called a node or point) represents an entity within a graph. The set of vertices in a graph G is denoted as *V(G)* or simply *V* when the context is clear [2].

c) **Edge (*E*):** An edge (also called a link or line) is a pair of vertices that specifies a connection between them. The set of edges in a graph G is denoted as *E(G)* or simply *E* [2].

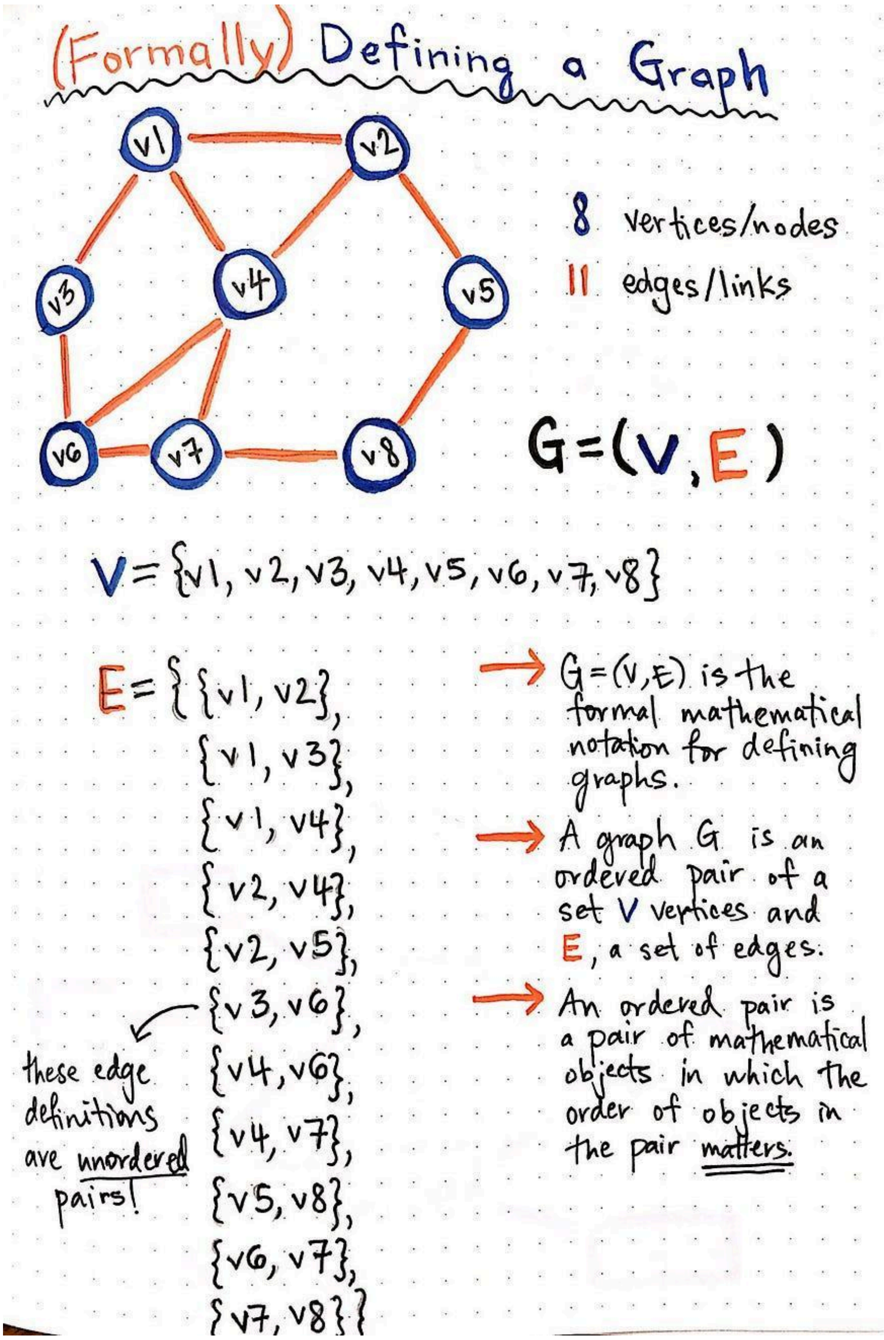

Fig. 2-An example of a simple graph (Joshi)

d) **Adjacent:** Two vertices are adjacent if they are connected by an edge. Similarly, two edges are adjacent if they share a common vertex [2].

e) **Degree of a Vertex:** The degree of a vertex in a graph is the number of edges incident to the vertex, with loops being counted twice [2].

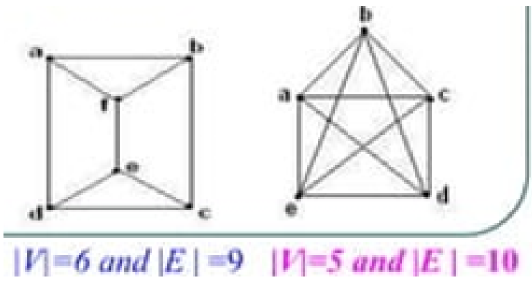

Fig. 3-Order and Size of a graph (Vainora and Boğaz)

f) **Magnitude and size of a graph-** The graph G is characterized by number of vertices $|V|$ (called the order of G) and the number of edges $|E|$ (size of G) [2].

g) **Path:** A path in a graph is a sequence of vertices where each adjacent pair is connected by an edge [2].

h) **Cycle:** A cycle is a path in which the start vertex and end vertex are the same, and no other vertices are repeated [2].

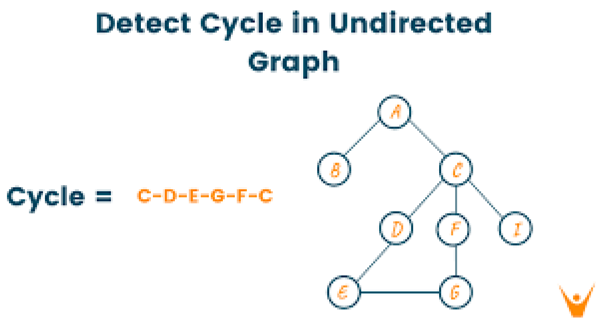

Fig. 4- Cycle of a simple undirected graph (Bhadaniya)

i) **Connected Graph:** A graph is connected if there is a path between every pair of vertices in the graph [2].

j) **Subgraph:** A subgraph is a graph whose vertex set and edge set are subsets of another graph [2].

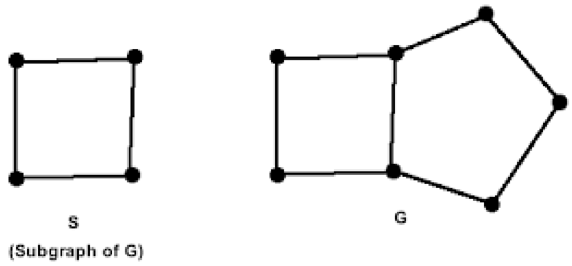

Fig. 5-Example of a subgraph of a graph (Unknown Author)

k) **Bipartite Graph:** A bipartite graph is a graph whose vertices can be divided into two disjoint sets such that every edge connects a vertex in one set to a vertex in the other set. The idea is to assign colors (or labels) to vertices during the traversal process and check if there are any adjacent vertices with the same color. If there are, then the graph is not bipartite. Otherwise, it is bipartite [2].

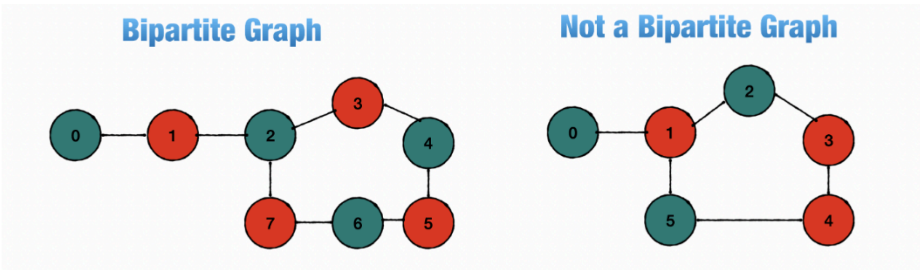

Fig. 6-Example of a Bipartite and non-Bipartite graph *(V)*

l) **Complete Graph:** A complete graph is a simple graph in which every pair of distinct vertices is connected by a unique edge [2].

m) **Directed Graph (Digraph):** A directed graph or digraph is a graph where the edges have a direction, indicated typically by arrows [2].

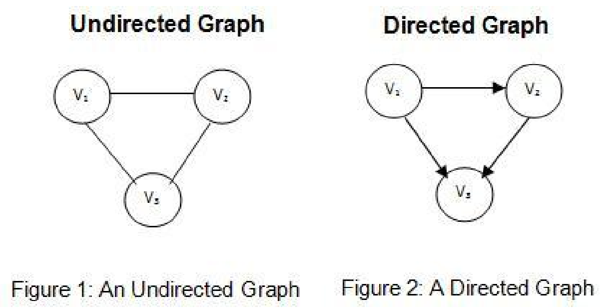

Fig. 7- Directed and Undirected Graph (Suman15)

4) Important theorem in Graph Theory (Euler's Theorem)

Euler's Theorem can be divided into two parts, concerning Eulerian paths and Eulerian circuits:

a) **Eulerian Path**: An Eulerian path in a graph is a path that visits every edge exactly once. Euler's Theorem states that a connected graph has an Eulerian path if and only if it has exactly 0 or 2 vertices of odd degree. If there are exactly 2 vertices of odd degree, the Eulerian path must start at one of them and end at the other [5].

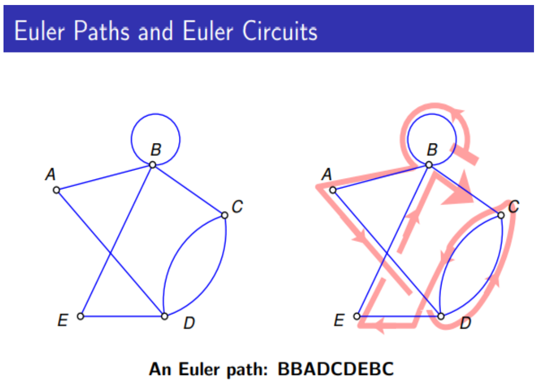

Fig. 8- Euler Path (Mahmood)

b) **Eulerian Circuit (or Eulerian Cycle):** An Eulerian circuit is an Eulerian path that starts and ends on the same vertex, thus forming a cycle. A connected graph has an Eulerian circuit if and only if every vertex has an even degree [5].

c) **Idea of proof for Eulerian Circuit**

- Necessity (if part):

Assume that a connected graph has an Eulerian circuit. Since it is a circuit, it visits every edge exactly once and returns to the starting vertex. Each time the circuit enters a vertex, it must leave the vertex. Therefore, the degree of each vertex must be even because it contributes 2 to the degree of each vertex: one for entering and one for leaving.

- Sufficiency (only if part):

Assume that every vertex in a connected graph has an even degree. Start traversing the graph from any vertex and keep moving until you return to the starting vertex. Since the graph is connected and every vertex has an even degree, whenever you enter a vertex, you must be able to leave it. This guarantees that you can traverse all edges without getting stuck. Eventually, you will return to the starting vertex because it has even degree, allowing you to leave it.

Thus, you have formed a circuit that visits every edge exactly once and returns to the starting vertex, fulfilling the definition of an Eulerian circuit.

In summary, the proof establishes that if and only if every vertex in a connected graph has an even degree, then the graph has an Eulerian circuit.

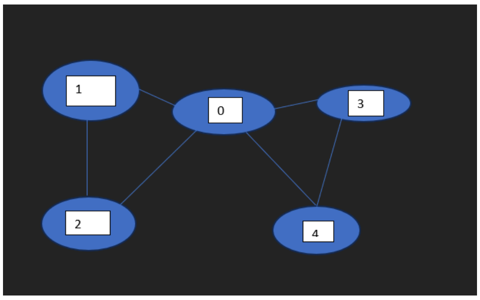

Fig. 9- A self-created example of an Eulerian Circuit

- Consider the example in fig. 9, Let's take a graph with 5 vertices, each having an even degree. In this graph:
- Each vertex has an even degree. It is connected, meaning there is a path between every pair of vertices. Now, let's trace a Eulerian circuit in this graph:
- 1 to 0, 0 to 3, 3 to 4, 4 to 0, 0 to 2, and finally 2 to 1.
- This path forms a Eulerian circuit since it visits every edge exactly once and returns to the starting vertex.
- So, in this example, the graph satisfies the condition that every vertex has an even degree, and it indeed has a Eulerian circuit.

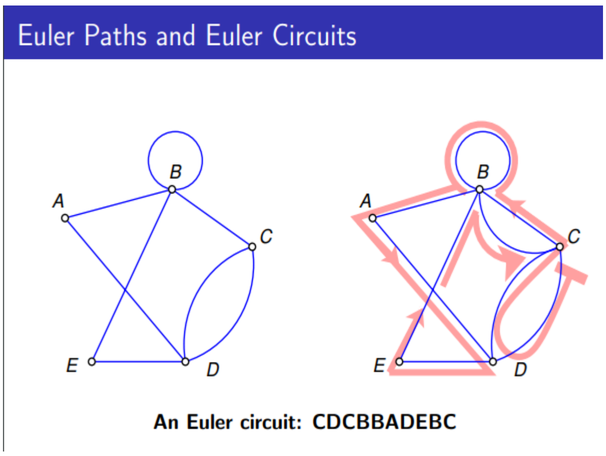

Fig.10- Another example of a Euler Circuit (Mahmood)

d) **Even vertex**- A vertex is considered an even vertex if its degree is an even number. This means that there is an even number of edges connected to it. For instance, if a vertex has 0, 2, 4, 6, etc., edges connected to it, it is classified as an even vertex. In practical terms, this means that for every edge leading into the vertex, there is another edge leading out.

MATH 341- Honors Project Spring 24 (Graph Theory)

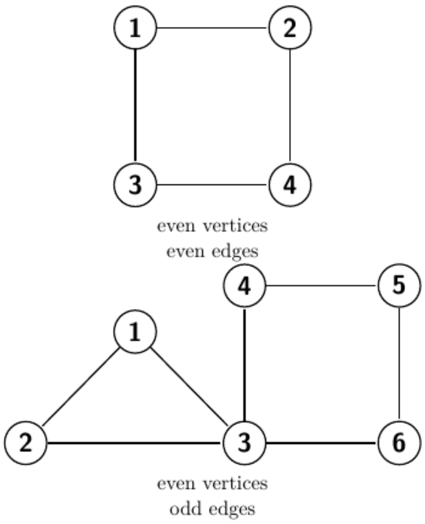

Fig.11- Even Vertices and Edges (Scott)

e) **<u>Odd Vertex</u>**: Conversely, a vertex is considered an odd vertex if its degree is an odd number. This means there is an odd number of edges connected to it, such as 1, 3, 5, 7, etc. An odd vertex indicates that there's an imbalance; for instance, if you were to try to enter and exit each vertex via a distinct edge (as in trying to find a Eulerian path or circuit), an odd vertex would either force you to start or end there because there would always be an extra edge unaccounted for in pairing.

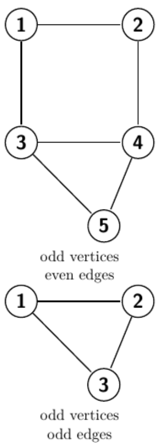

Fig. 12- Odd Vertices and Edges (Scott)

5) Application of Graph Theory in The PageRank Algorithm

MATH 341- Honors Project Spring 24 (Graph Theory)

The PageRank algorithm is a method developed by Larry Page and Sergey Brin, the founders of Google, to rank web pages in search engine results. It was a foundational algorithm that helped set Google apart in the early days of the web by determining the importance of website pages based on their link structure. The core idea behind PageRank is that the significance of a web page can be determined by the number and quality of links pointing to it. In other words, a page receives a higher rank if it is linked to by other pages that are themselves important.

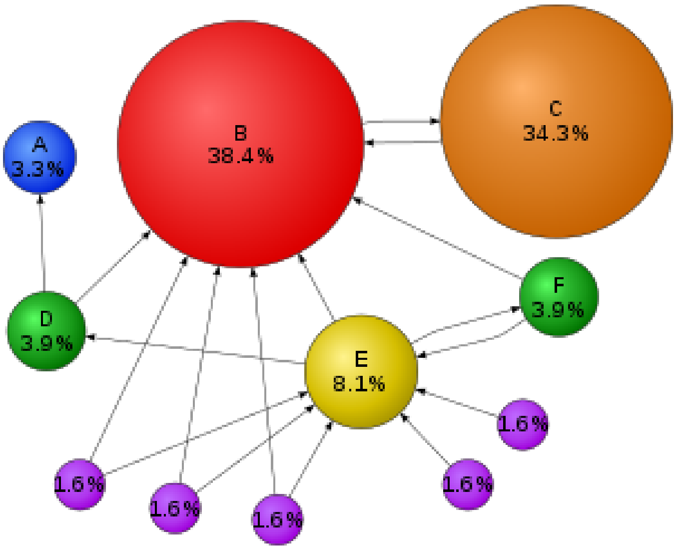

Fig. 13-PageRank Algorithm (Tekula)

PageRank directly applies graph theory by treating the World Wide Web as a directed graph to calculate the importance of each webpage. Here's how the graph theory concepts are utilized in PageRank:

a) **<u>Vertices (Nodes)</u>**: Each webpage on the internet is represented as a vertex (or node) in the graph.

b) **<u>Edges (Links):</u>** Each hyperlink from one webpage to another is represented as a directed edge from the corresponding vertex of the first webpage to the vertex of the second. The direction indicates the link direction, from the linking page to the linked page.

c) Application of Graph Theory in PageRank

- **Graph Representation**: The entire web can be visualized as a vast directed graph where each page (node) is connected to others through links (directed edges). This graph is not static but constantly evolves as new pages are added and old ones are removed, and as links are created and broken [1].

- **Link Analysis**: PageRank analyses the connectivity of the graph to determine the importance of each node. In graph-theoretic terms, it examines how vertices are connected and uses the structure of these connections to assign a weight to each vertex [1].

- **Rank Calculation**: The fundamental idea behind PageRank is that a link from page A to page B can be seen as an endorsement of page B by page A. The more endorsements (incoming links) a page receives, the more important it is. However, not all endorsements are equal – an endorsement from a highly important page is worth more than one from a lesser-known page. This concept is captured using the adjacency matrix of the graph, where each cell in the matrix represents the presence of an edge (link) from one node to another, and calculations are performed iteratively to distribute ranks through the network.

- **Damping Factor**: The damping factor in PageRank mimics the behaviour of a web surfer who randomly clicks on links but occasionally jumps to a new page not directly linked to the current one. This behaviour introduces the concept of random walks on the graph, where the probability of moving from one node to another is influenced by the structure of the graph itself [1].

- **Convergence through Iteration**: The iterative process of updating PageRank scores based on the inbound links and their own PageRank scores reflects the idea of spreading activation in a graph. The process continues until the scores stabilize (converge), which means that the graph has reached an equilibrium state where the importance of each node accurately reflects its position and connectivity within the network.

6) Conclusion

In conclusion, graph theory represents a fascinating and dynamic branch of mathematics, offering a framework for understanding and analyzing the interconnectedness of diverse systems through the abstraction of vertices and edges. From its inception with Euler's exploration of the Seven Bridges of Königsberg, graph theory has evolved into a rich field with broad applications in mathematics and beyond. Its fundamental principles and theorems continue to inspire exploration and innovation, promising further insights into the complexities of networks and paving the way for future discoveries.

MATH 341- Honors Project Spring 24 (Graph Theory)

Works Cited

1) Amine, Amrani. "PageRank Algorithm, Fully Explained." *Medium*, Towards Data Science, 20 Dec. 2020, towardsdatascience.com/pagerank-algorithm-fully-explained-dc794184b4af.

2) Balakrishnan, R., and K. Ranganathan. *A Textbook of Graph Theory*. 2nd ed., Springer, 2012.

3) Bhadaniya, Shivali. "Detect Cycle in an Undirected Graph Using DFS (with Code)." *FavTutor*, FavTutor, 22 June 2023, favtutor.com/blogs/detect-cycle-in-undirected-graph.

4) Carlson, Stephan C. "Graph Theory." *Encyclopædia Britannica*, Encyclopædia Britannica, inc., 29 Feb. 2024, www.britannica.com/topic/graph-theory.

5) "Euler Graph in Discrete Mathematics - Javatpoint." *Www.Javatpoint.Com*, www.javatpoint.com/euler-graph-in-discrete-mathematics. Accessed 13 Mar. 2024.

6) "Graphs and Subgraphs." *MathOnline*, WikiDot.com, mathonline.wikidot.com/graphs-and-subgraphs. Accessed 2 Apr. 2024.

7) "Graph Theory." *Wikipedia*, Wikimedia Foundation, 14 Feb. 2024, en.wikipedia.org/wiki/Graph_theory#:~:text=In%20mathematics%2C%20graph%20theory%20is,also%20called%20links%20or%20lines).

8) Joshi, Vaidehi. "An Example of a Simple Graph." *Medium*, basecs, 25 May 2017, medium.com/basecs/a-gentle-introduction-to-graph-theory-77969829ead8.

9) Mahmood, Omer. "Genetics-Towards Data Science." *Euler Paths and Euler Circuits*, Towards Data Science (TDS), 21 Oct. 2022, towardsdatascience.com/tagged/genetics.

10) Scott, Brian M. *Eulerian Graph with Odd/Even Vertices/Edges*, StackExchange, 1 Aug. 1961, math.stackexchange.com/questions/1495860/eulerian-graph-with-odd-even-vertices-edges.

11) Suman15. "OSPF (Open Short Path First) Routing Protocol Implemented Using Dijkastra Algorithm." *OSPF (Open Short Path First) Routing Protocol Implemented Using Dijkastra Algorithm*, Medium, 8 Aug. 2021, suman15.medium.com/ospf-open-short-path-first-routing-protocol-implemented-using-dijkastra-algorithm-7893d4e55683.

12) Tekula, Mike. "Beginner's Guide to Link Metrics." *Search Engine Journal*, Search Engine Journal, 19 Oct. 2010, www.searchenginejournal.com/beginners-guide-to-link-metrics/19926/.

13) V, Rohith. "Graph Algorithm - Bipartite Graph(Bfs)." *DEV Community*, DEV Community, 5 Apr. 2022, dev.to/rohithv07/graph-algorithm-bipartite-graphbfs-4f3b.

14) Vainora, Julius, and Semih Boğaz. "Graph Theory, Wheel Graphs." *Math- Graph Theory, Wheel Graphs*, Stack Overflow, 1 Oct. 1964, stackoverflow.com/questions/53899887/graph-theory-wheel-graphs.