

Jñānābha, Vol. 56(1P) (2026), 1-8

GRAPH THEORY FUNDAMENTALS AND APPLICATIONS

Riddhi Jangid, Gajendra Pratap Singh and Tushar Dedha*

School of Computational and Integrative Sciences Jawaharlal Nehru University, New Delhi 110067

Email: riddhijangid97@gmail.com, gajendra@jnu.ac.in, tushardedhajnu@gmail.com

*Corresponding Author

(Received: August 01, 2025, Revised: February 15, 2026; Accepted: March 10, 2026)

DOI: <https://doi.org/10.58250/jnanabha.2026.561P-1>

Abstract

Graphs are mathematical structures composed of nodes and edges representing entity relationships. They help analyze the network properties, and model real-world phenomena to understand the complex and tricky systems. There are several applications of graphs in real-life problems. This article deals with the applications and modeling of graph theory in computer science, network biology, and chemical processes. We can also use graph theory to study the structural properties of network topologies such as star and wheel graphs, and Peterson graphs. It can also be used to learn the graph operations that can help study the infectious diseases network like COVID-19. Also, it can study network optimization by finding the optimal cost, optimal route, shortest distance, and the study of spanning networks. This is a fundamental article specifically for beginners.

2020 Mathematical Sciences Classification: 05Cxx, 05C85, 05C90

Keywords and Phrases: Graphs, networks, adjacency matrix, modelling, complex systems.

1 Introduction

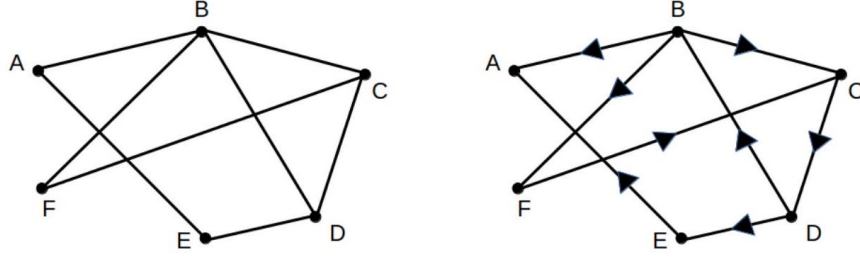
A famous Knigsberg problem was solved by a Swiss mathematician, Leonhard Euler in 1736 and since then he became the father of Graph Theory. In 1847, the theory of trees was introduced by Kirchhoff for solving a system of linear equations. Graphs are mathematical structures composed of nodes and edges representing entity relationships [5]. These nodes and edges can be used to create a mathematical model for solving complex and tricky systems [12]. They are also used to represent multiple types of networks in different domains of science like communication networks, biological networks [7], computer networks, etc. In the next section, we introduce the preliminaries of graph theory that are general and can be used in any type of modeling. In sections 3 and 4, we introduce a new example of graph theoretical algorithms called Breadth First Search and Depth First Search. In section 5 we've explained a basic example of knights moves on a 3×3 grid using graph modeling and conclude with a discussion of the introduced concepts.

2 Preliminaries

We discuss here the fundamental definitions of graphs in the section. These definitions lay the foundation of what we discuss in the further sections.

2.1 Graph

A graph G is a diagrammatic representation of showing relationships/connections between entities. Entities are denoted using nodes, V and relationships among them are depicted using edges, E between the nodes. Mathematically, $G := (V, E)$ where V and E are a finite set of nodes and edges. It is important to note that V must be non-empty. E in G are of two types, directed and undirected. Fig. 2.1(a) shows an undirected graph with 6 nodes and 8 edges and Fig. 2.1(b) shows a directed graph with 6 nodes and 8 edges.



Example of (a) An Undirected Graph

(b) Directed Graph

Figure 2.1: Example of (a) An Undirected Graph (b) Directed Graph

2.2 Loop

A loop is an edge of a graph whose starting and ending nodes are the same. While considering the definition of a simple graph, a loop is not allowed.

2.3 Degree

A degree is associated with the nodes of a graph. For an undirected graph, the number of edges associated with a vertex is called the degree of that vertex. But in a directed graph, every vertex has an in-degree and an out-degree which is defined as the number of incoming edges and the number of outgoing edges to any vertex. For example, vertex B in Fig. 2.1(a) has degree 4 and vertex B in Fig. 2.1(b) has indegree 1 and out-degree 3.

2.4 Handshaking lemma

The Handshaking lemma states that the sum of the degrees of all the vertices in an undirected graph is always equal to twice the number of edges in the graph.

2.5 Adjacency Representation

We represent a simple graph using an Adjacency Matrix. The matrix is such that all of its entries are Boolean and order is $n \times n$ if n is the number of vertices of the graph. We assign value 1 to a_{ij} entry of an adjacency matrix A if there is an edge between the i^{th} and j^{th} node of the graph. For eg. if we represent the graphs in Fig. 2.1. Using the Adjacency matrix then they can be represented as follows.:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

represents adjacency matrix for undirected graph (see Fig 2.1(a)) and

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

represents the adjacency matrix for the directed graph (see Fig 2.1(b)).

2.6 Walk

In a graph, the walk is defined as a sequence containing edges and vertices alternatively. It is like we are walking on a graph. In walk, vertices' and edges can be repeat any number of times. For eg., $C \rightarrow B \rightarrow F \rightarrow C \rightarrow D \rightarrow B \rightarrow C$ is a walk of Fig. 2.1(a).

2.7 Trail

The trail is a walk such that we move on an edge only once. For eg., In Fig. 2.1(a) $C \rightarrow B \rightarrow F \rightarrow C \rightarrow D$ is an open trail while $A \rightarrow B \rightarrow D \rightarrow E \rightarrow A$ is a closed trail. A closed trail is also called a circuit.

2.8 Path

A Path in a graph is defined as a walk in which we neither repeat a vertex nor an edge. For eg., $B \rightarrow F \rightarrow C \rightarrow D \rightarrow E \rightarrow A$ is a path in Fig. 2.1(a).

2.9 Tree

We call a connected graph a tree when there is no cycle present in an undirected graph. In other words, we say when any two vertices are connected by a single edge only in a graph then it is a tree.

2.10 Eulerian Graph

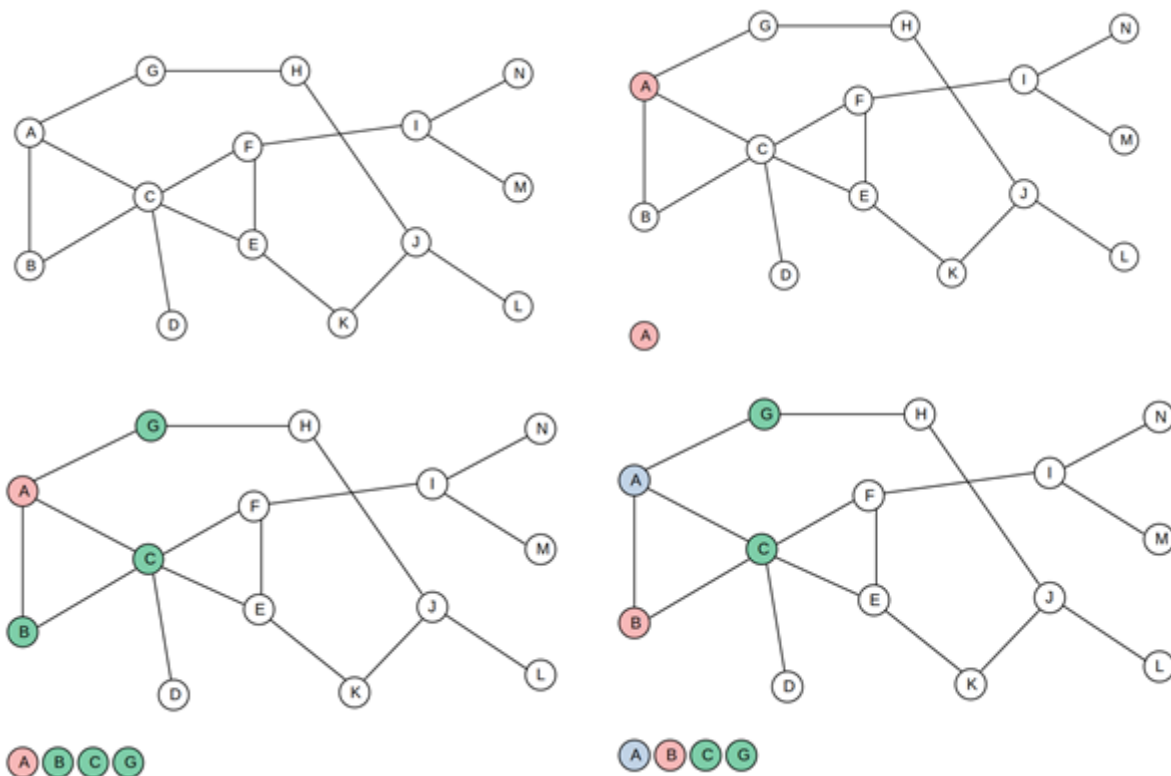
A connected graph on which we can walk on all the edges without repeating any edge and come back to the same vertex where we started then that type of graph is called the Eulerian graph. For eg., Fig. 2.1(a) is not an Eulerian Graph.

2.11 Hamiltonian Graph

A connected graph on which we can walk on all the vertices without repeating any vertex and come back to the same vertex where we started then that type of graph is called a Hamiltonian graph. For e.g., we have Hamiltonian circuit $B \rightarrow F \rightarrow C \rightarrow D \rightarrow E \rightarrow A \rightarrow B$ in Fig. 2.1(a).

3 Breadth First Search Algorithm

Breadth First search is a graph traversal technique that is used to explore nodes and edges of a graph. It is used as a fundamental algorithm for many tasks like shortest path finding, cycle detection, topological sorting, etc. It begins at a vertex and covers all the nodes by covering the neighborhood of nodes at each level. An example is shown below to explain the steps of the algorithm.



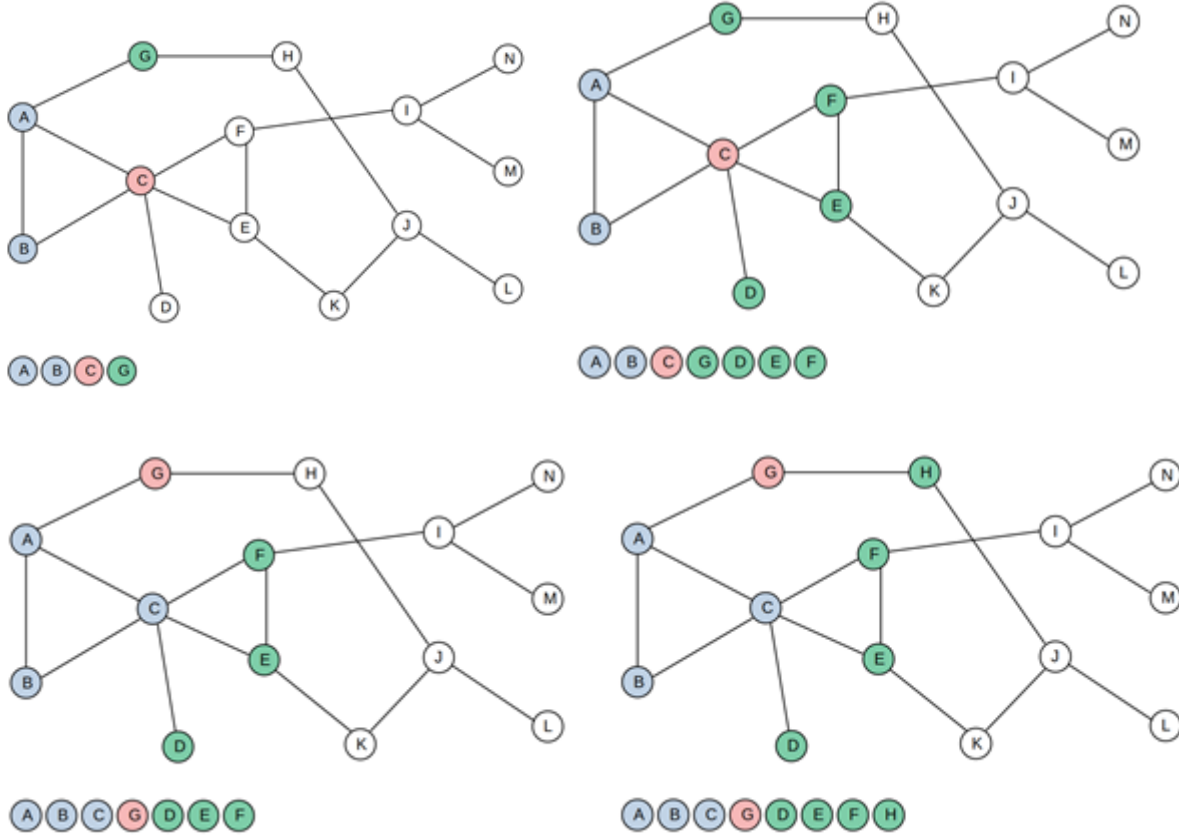


Figure 3.1: (a) (b) (c) (d) (e) (f) (g) (h) An illustrative example of Depth First Search Algorithm

In Fig. 3.1, an example of a graph with 14 vertices and 16 edges is considered. We apply the Breadth First Search on the graph and a few initial steps have been explained in the figure. The initial vertex chosen to start from is shown in red color. Suppose, we start with node A and look for its neighbors that are depicted using the green color of the nodes. After covering all the neighbors of a node and adding them to the queue we color it with blue, move on to the next vertex of the queue, and repeat the process using the next red vertex. Proceeding in the same way, we have the following sequence of vertices as our result of the algorithm.



Figure 3.2: Solution to the Breadth First Search of the example shown in Fig. 3.1.

4 Depth First Search

It is another graph traversal technique used to explore every node of a graph or a tree. We start with an initial vertex and explore each vertex as long as possible along every branch and then move backward after reaching the end of the branch. We illustrate the algorithm in Fig. 4.1. with the help of the same graph used to explain the Breadth First Search Algorithm.

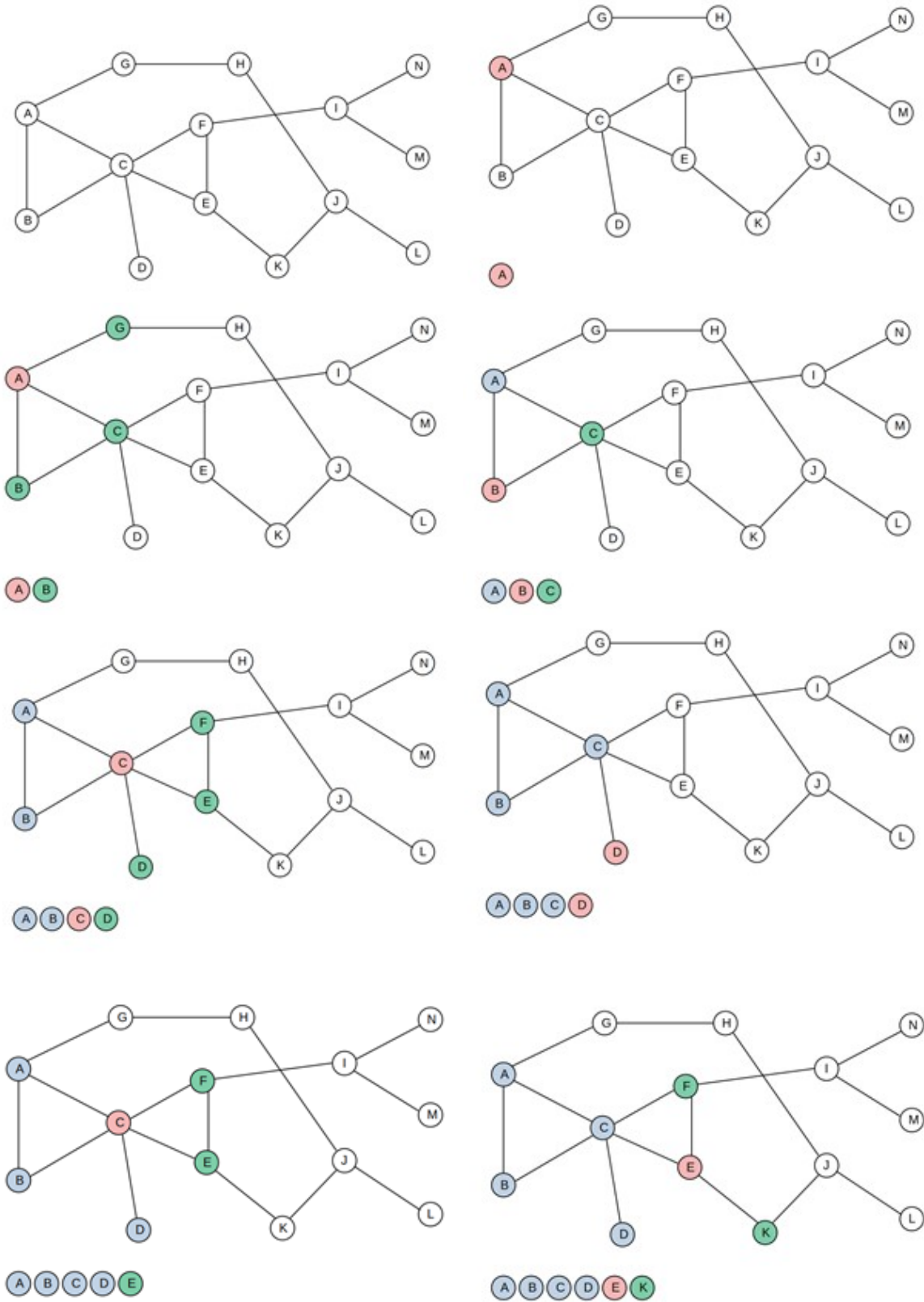


Figure 4.1: (a) (b) (c) (d) (e) (f) (g) (h) An illustrative example of Depth First Search Algorithm

The algorithm continues and when all the vertices are covered then we have the following sequence of vertices shown in Fig. 4.2.



Figure 4.2: Solution to the Depth First Search of the example shown in Fig. 4.1.

5 Applications of Graphs

Graph theory is used in many ways [3]. Theoretically, while studying communication networks, biological networks, computer networks, etc. we use graphs to study the topological properties of these networks. In real-life problems [10], we use graph theory, like in Airline scheduling, and directions in Google Maps, Sudoku's puzzles, Social Media marketing, etc. These problems use different graph theory topics like graph coloring, shortest path problems, etc [1,6].

We illustrate the application of graphs using an example. A very interesting example is studying the knight's moves on a chess board. Consider we have a 3×3 grid and we wish to understand the minimum number of moves required from the initial position to the final position of the 4 knights (2 white and 2 black) as shown in the Fig. 5.1.

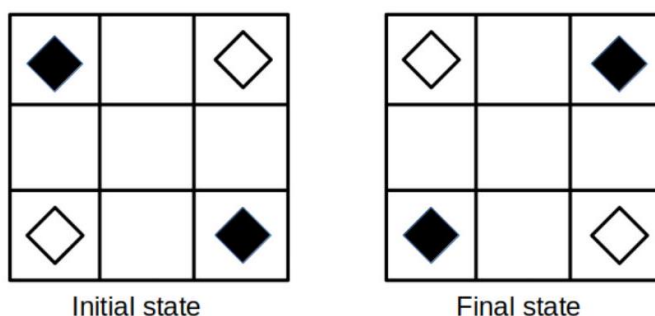


Figure 5.1: Example of knight moves on a 3×3 grid

First question to consider is, Is this final state possible? If yes, then what is the minimum number of steps required to reach final state? Solution to these questions can be hit and trial method but this situation can be tricky when we have a larger grid. So we use graph to make a understand the positions of the knights in the grid. Let us number these tiles from 1-9 on the grid as shown in the Fig. 5.2.

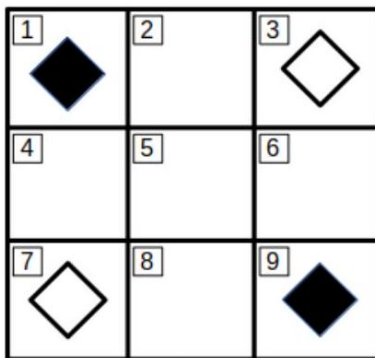


Figure 5.2: Numbering the tiles of the 3×3 grid

We make a graph of the moves that these 4 knights can move and understand the dynamics using the graph. This can be illustrated in the Fig. 5.3. We can clearly see the connections between these vertices that depicts the possible moves of the four knights on the grid.

To understand the possible moves and their minimum number of steps. We count the steps on the graph and that will be our required solution. Fig. 5.4. Depicts that minimum number of moves required to reach from initial to final state is 8. We can model more such examples

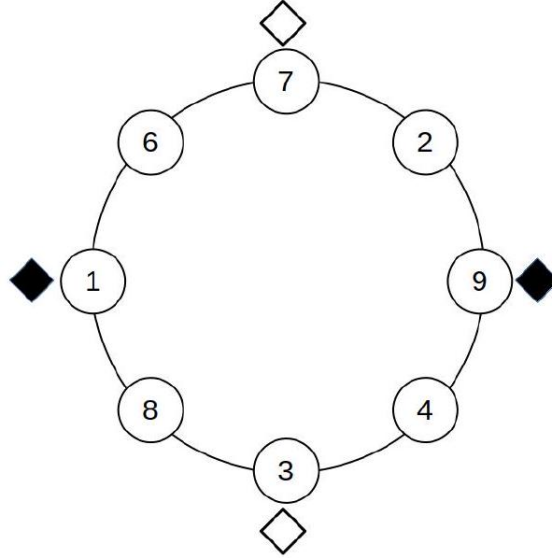


Figure 5.3: Graph of all the possible knights move on the grid

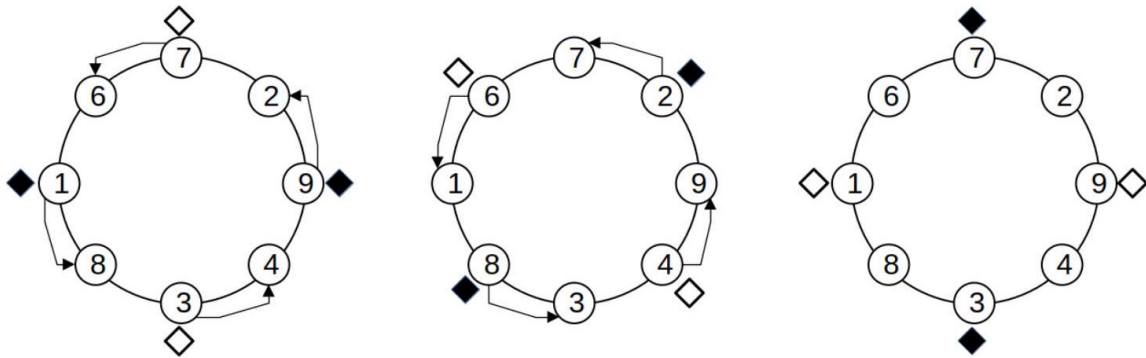


Figure 5.4: Knights move to reach from initial to final state

6 Conclusion and Discussion

Graph theory has gained a lot of attention due to its applicability and modeling power. It is vastly used in many areas of sciences. Types of graph like bipartite graph, complete graph, etc. have proven to be useful in different applications. Extension to graph theory have been studied extensively like Petri Nets [1, 2, 4, 5, 8, 9, 11, 13-19] and Signed Graphs. Petri Net is the modeling tool used to study the behavior of discrete event type dynamic systems. Signed graphs extends the modeling power of graphs by introducing the positive and negative sign to the edges of the graph. We have discussed the fundamentals of graph in this paper and readers are motivated to apply these concepts in their area of study.

Acknowledgement. The second author, Dr. Gajendra Pratap Singh is thankful to the Department of Science and Technology(DST)- Science and Engineering Research Board(SERB) Projects (PID:M

TR/2021/000378), and the authors are expressing their deep gratitude to anonymous reviewers and editors for their valuable suggestions and comments.

References

- [1] A. T. Balaban, Applications of graph theory in chemistry, *Journal of Chemical Information and Computer Sciences*, **25**(3) (1985), 334–343.
- [2] A. Borah and G. P. Singh, Some Generalizations of Corona Product of Two Graphs, *Applications and Applied Mathematics: An International Journal (AAM)*, **19**(3) (2024), 3.
- [3] J. L. Gross, J. Yellen and M. Anderson, *Graph theory and its applications*, Chapman and Hall/CRC (2018).
- [4] S. Gupta, G. P. Singh and S. Kumawat, Modeling and targeting an essential metabolic pathway of *Plasmodium falciparum* in apicoplast using Petri nets, *Applied Mathematics-A Journal of Chinese Universities*, **37**(1) (2022), 91–110.
- [5] F. Harary, *Graph theory (on Demand Printing of 02787)*, CRC Press, 2018.
- [6] L. J. Hubert, Some applications of graph theory to clustering, *Psychometrika*, **39**(3) (1974), 283–309.
- [7] R. Jangid, P. Somvanshi and G. P. Singh, Graph Theory in the Biological Networks, in *Biological Networks in Human Health and Disease*, Springer Nature Singapore (2023), 1–14.
- [8] R. Jangid and G. P. Singh, Construction of m-cube mesh networks using Petri nets, *International Journal of System Assurance Engineering and Management*, **14**(Suppl 1) (2023), 256–262.
- [9] M. Jha, M. Singh and G. P. Singh, Modeling of second-line drug behavior in the treatment of tuberculosis using Petri net, *International Journal of System Assurance Engineering and Management*, (2022), 1–10.
- [10] M. A. Kanhio, Scope of Graph Theory in Real Life, *Available at SSRN 4651239*, (2023).
- [11] S. Kansal, G. P. Singh and M. Acharya, On the problem of characterizing Boolean Petri nets, *International Journal of Computer Applications*, **975** (2015), 8887.
- [12] A. Prathik, K. Uma and J. Anuradha, An Overview of application of Graph theory, *International Journal of ChemTech Research*, **9**(2) (2016), 242–248.
- [13] S. K. Singh, R. Jangid and G. P. Singh, On characterizing binary Petri Nets, *International Journal of System Assurance Engineering and Management*, **14**(3) (2023), 919–929.
- [14] G. P. Singh, A. Borah and S. Ray, A review paper on corona product of graphs, *Advances and Applications in Mathematical Sciences*, **19**(10) (2020), 1047–1054.
- [15] G. P. Singh, S. K. Singh and M. Jha, Existence of forbidden digraphs for crisp Boolean Petri nets, *International Journal of Mathematical, Engineering and Management Sciences*, **5**(1) (2020), 83.
- [16] G. P. Singh, M. Jha, M. Singh and Naina, Modeling the mechanism pathways of first line drug in Tuberculosis using Petri nets, *International Journal of System Assurance Engineering and Management*, **11** (2020), 313–324.
- [17] G. P. Singh and S. K. Singh, Petri net recommender system for generating of perfect binary tree, *International Journal of Knowledge and Systems Science (IJKSS)*, **10**(2) (2019), 1–12.
- [18] G. P. Singh and A. Gupta, A Petri net analysis to study the effects of diabetes on cardiovascular diseases, in *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE (2019), 481–488.
- [19] G. P. Singh, A Wheel 1-Safe Petri Net Generating all the $\{0,1\}^n$ Sequences, *International Journal of Computer Applications*, **84**(16) (2013).
- [20] R. N. Verma, M. Z. Malik, N. Subbarao, G. P. Singh and D. N. Sinha, Entamoeba histolytica HM-1: IMSS gene expression profiling identifies key hub genes, potential biomarkers, and pathways in Amoebiasis infection: A systematic network meta-analysis, *Bioscience Reports*, **42**(10) (2022), BSR20220191.
- [21] R. N. Verma, M. Z. Malik, G. P. Singh and N. Subbarao, Identification of key proteins in host-pathogen interactions between *Mycobacterium tuberculosis* and *Homo sapiens*: A systematic network theoretical approach, *Healthcare Analytics*, **2** (2022), 100052.