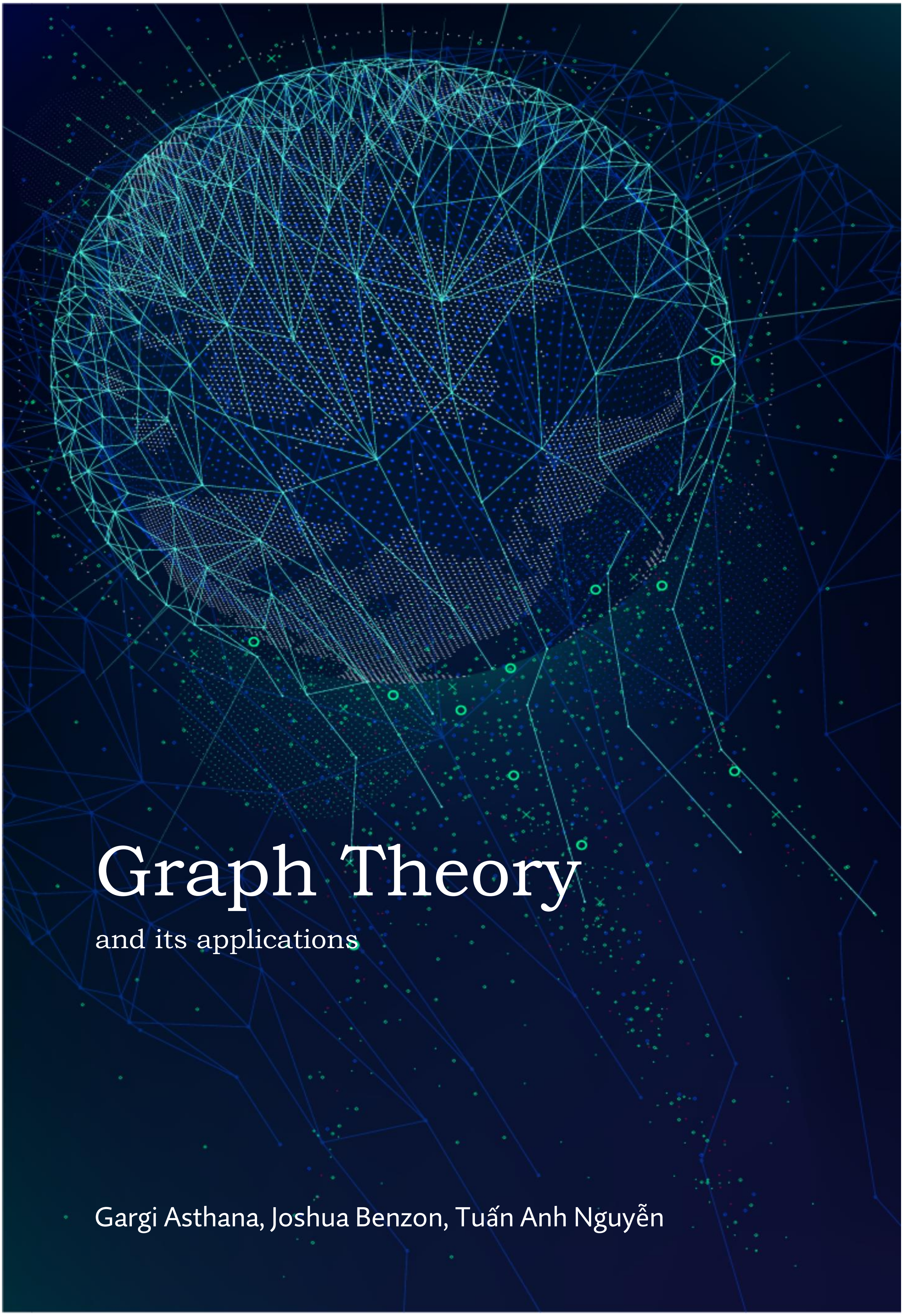

Graph Theory

and its applications

Gargi Asthana, Joshua Benzon, Tuấn Anh Nguyễn

Table of Contents

INTRODUCTION:

Graph Theory is the study of graphs or mathematical structures used to model pairwise relations between two objects. It is a branch of discrete mathematics distinguished by its geometric approach to study various objects.
A graph connects two objects, these objects are known as vertices (also known as nodes or points). The vertices are connected by edges (also known as links or lines).

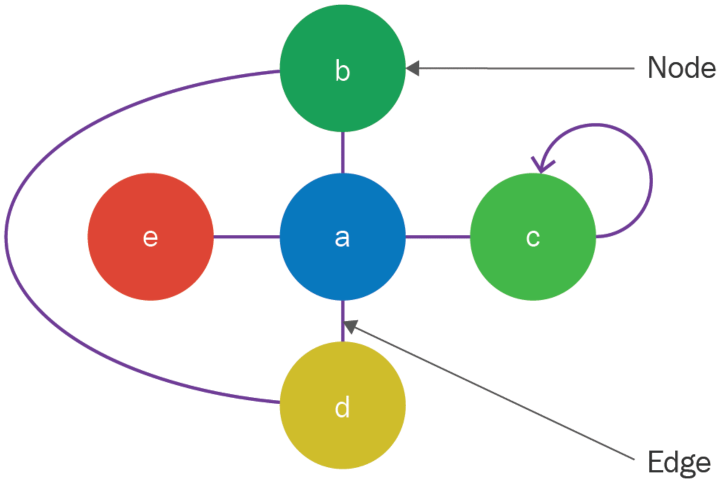

Figure 1.1

There are mainly two types of graphs – Directed and Undirected Graphs. However, both are further classified as Simple Graphs or Multigraphs.

1. **Undirected Graphs**
   a. Undirected Simple Graph

   A graph can be represented by an ordered pair:

$$G = (V, E)$$

   Where:

   - V is a set of vertices
   - $E \subseteq \{\{x, y\} \,|\, x, y \in V \text{ and } x \neq y\}$

     represents a set of edges, which are unordered pairs of vertices.

   A simple graph does not contain multiple edges.

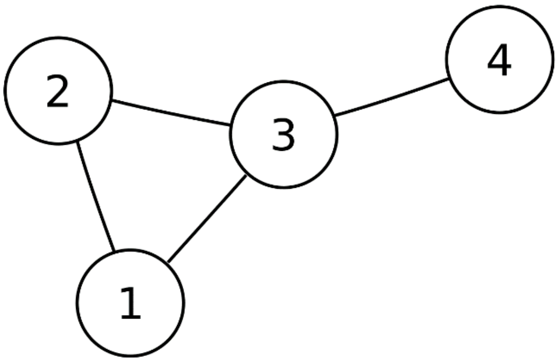

Figure 1.2

b. Undirected Multigraph

A graph can be represented by an ordered pair:

$$G = (V, E, \emptyset\,)$$

Where:

- V is a set of vertices
- E is a set of edges
- $\emptyset: E \to \{(x, y) |\ (x, y) \in V^2\ and\ x \neq y\}$

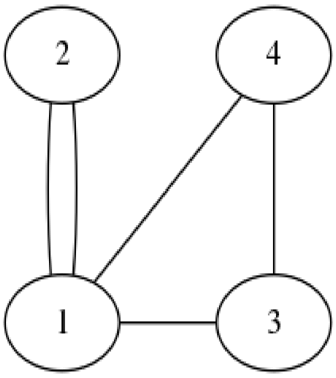

Figure 1.3

2. Directed Graphs (Digraphs)

a. Directed Simple Graph

A digraph is also represented by an ordered pair:

$$G = (V, E)$$

Where:

- V is a set of vertices
- $E \subseteq \{\{x, y\} \,|(x, y) \epsilon\ V^2 and\ x \neq y\}$
  represents a set of edges (also known as directed edges, directed links, arrows or arcs), which are ordered pairs of vertices.

In the edge (x, y) directed from x to y, the vertices are called endpoints of the edge. Here, x is the tail of the edge whereas y is the head of the edge.
The edge (y, x) is called the inverted edge of (x, y).
Multiple edges are not allowed in such a case.

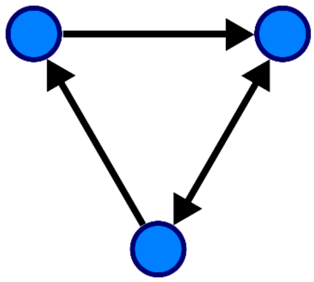

Figure 1.4

b. Directed Multigraph

A directed graph has an ordered triple:

$$G = (V, E, \emptyset)$$

Where:

- V is a set of vertices
- E is a set of edges
- $\emptyset: E \to \{(x, y) | (x, y) \in V^2 \; and \; x \neq y\}$

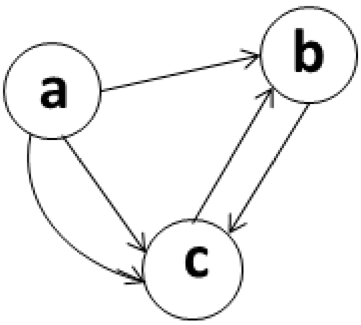

Figure 1.5

Important Terminology:

Walk: Any random traversal in a graph where both vertices and edges can be repeated.

Trail: A walk in a Graph in which no edge can be repeated however, vertices may be repeated.

EULER GRAPH AND CIRCUIT

History

The *Seven Bridges of Königsberg* is a historically notable problem in mathematics. It would be very difficult to find Königsberg on any modern maps, but one special peculiarity in its geography has made it one of the most famous cities in Mathematics. The city of Königsberg in Prussia (now Kaliningrad, Russia) laid on both sides of the Pregel River and included two large islands. The two islands were connected to each other and to the riverbanks by seven bridges.

Carl Gollieb Ehler, a mathematician who later became a mayor in a nearby town, was absolutely obsessed with the islands and bridges. There was a single question which was always racing through his mind: “Which route would allow him to cross all seven bridges by crossing any of them exactly once?” The final answer was “impossible”.

Carl wrote for Leonhard Euler, who was a very famous mathematician, for help with this problem. Fortunately, that was the beginning for a very new field of Mathematics. Euler initially dismissed this question as having no interest in it. But the more he wrestled with it, he gradually felt there would be something exciting after all. The answer he came up with was a type of geometry which did not quite exist before, what he called Geometry of Position, known as *Graph Theory*, nowadays.

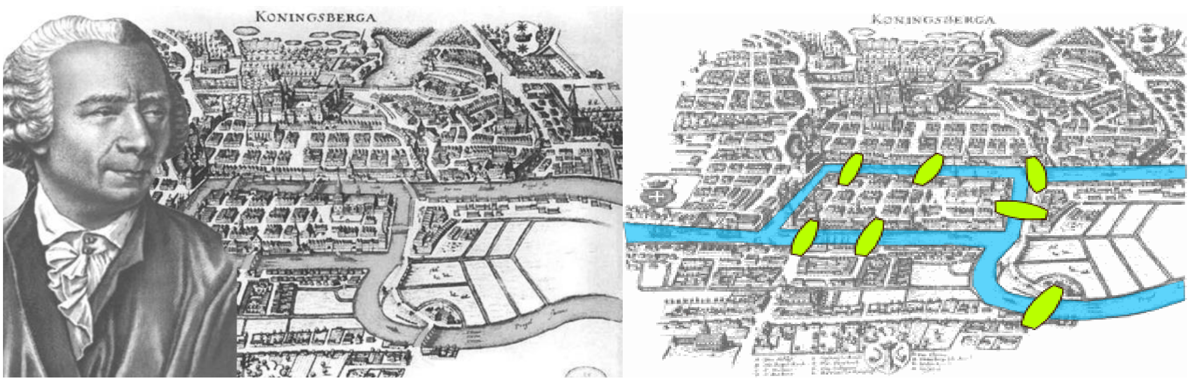

Figure 2.1. ***Map of Königsberg in Euler's time showing the actual layout of the seven bridges, highlighting the river Pregel and the bridges***

Theory

A. Euler Path and Euler Circuit

- The map in Figure 1 could be simplified with each of the four areas represented as a single point, what we now call a node, or a vertex with lines, or edges between them to represent the bridges. (As you can see in Figure 2.2)

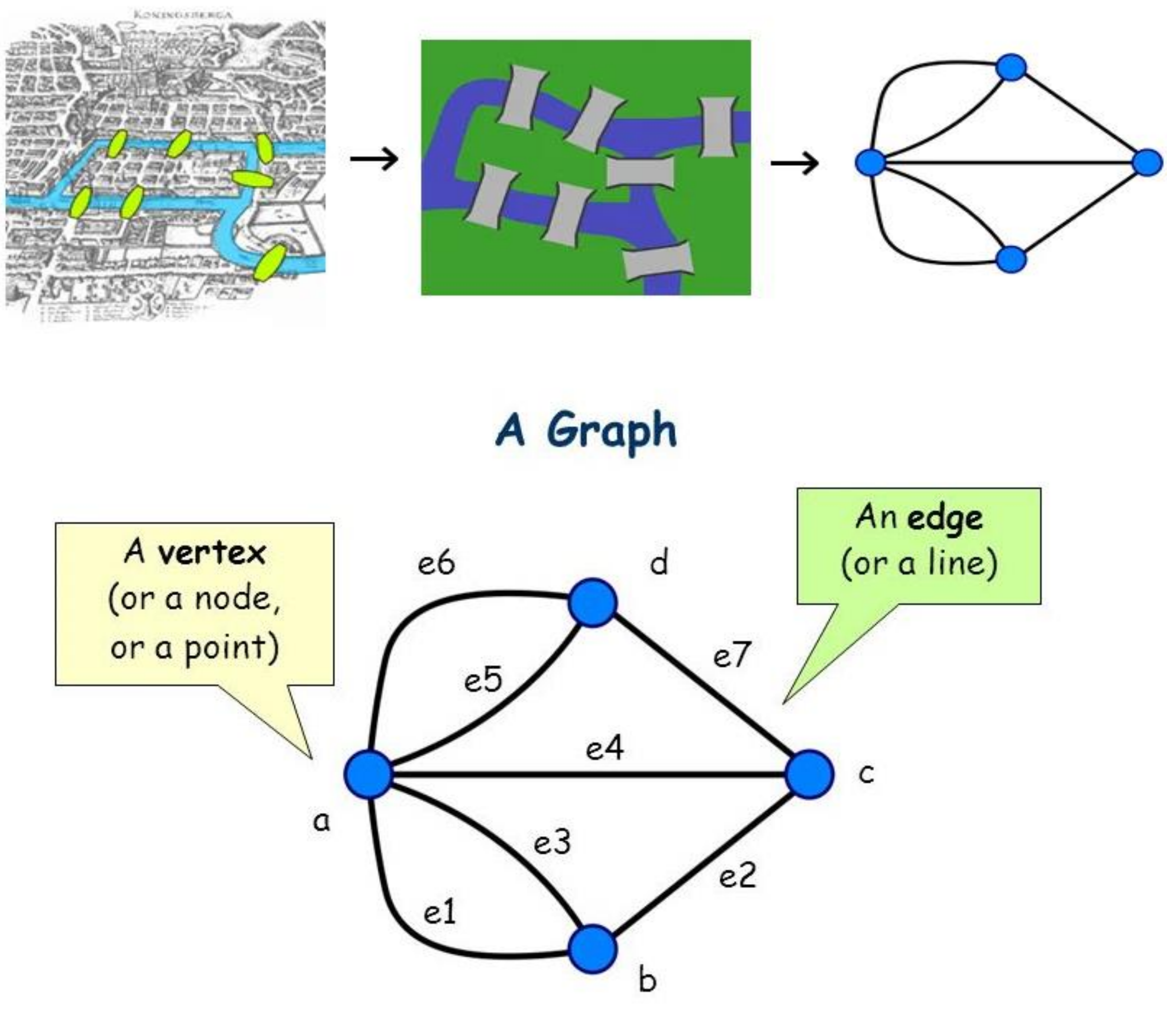

Figure 2.2. ***Seven Bridges of Königsberg Puzzles***

- **<u>Definition 2.1</u>:** A *Euler Path* "walks" through a graph using every edge exactly once. Being a path, it does not have to return to the starting vertex. (Figure 2.3)

- **<u>Definition 2.2</u>:** A *Euler Circuit* has the condition of an *Euler Path*, additionally starts and stops at the same vertex. (Figure 2.3)

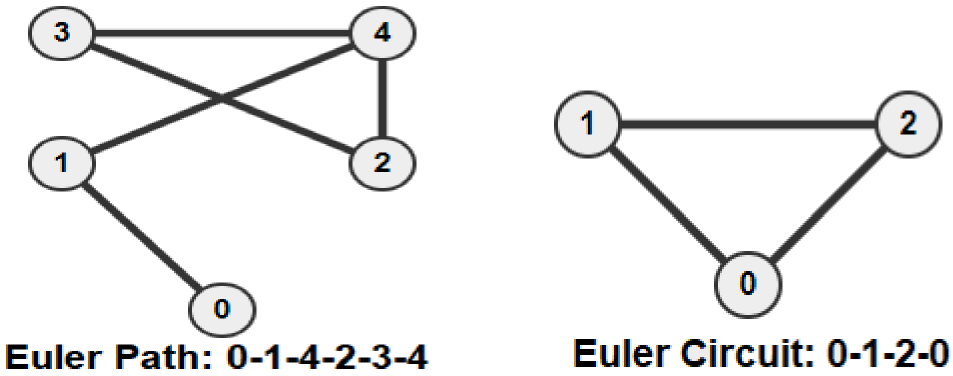

Figure 2.3. ***Euler Path and Euler Circuit***

B. <u>Euler Path and Euler Circuit Theorems</u>

- **<u>Theorem 1:</u>** A graph will contain a *Euler Circuit* if all vertices have even degrees.

- **<u>Theorem 2</u>:** A graph has a *Euler Path* but not a Euler circuit if and only if it has exactly two vertices of odd degrees.

<u>Proof</u>: For theorem 1, assume the graph has a Euler path but not a circuit. Notice that every time the route passes through a vertex, it adds 2 to the degree of the vertex (one for entering and one for leaving). Apparently, the first and the last vertices will have odd degrees and all the other vertices are even degrees.

For theorem 2, it is like theorem 1, but the path of a circuit starts and stops at the same vertex. Therefore, the first and the last vertices will also have even degrees. Generally, all vertices will have even degrees

- **<u>Note</u>**: The *degree* of a vertex is the number of edges meeting at that vertex.

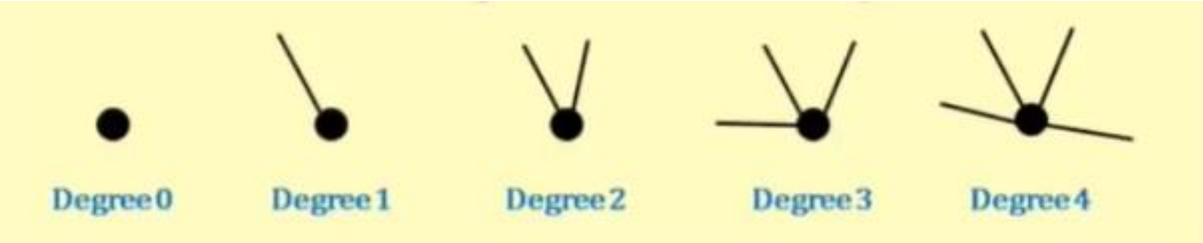

Figure 2.4. ***Degree of a vertex***

Euler used this proof to formulate a general theory which applies to all graphs with two or more nodes. This theory has totally answered the question of Carl. Since the bridges of Königsberg graph had all four vertices with odd degree, there is no Euler path through the graph. Thus, there is no way for him to cross every bridge exactly once. However, how might you create a Euler path in Königsberg? The answer will surely be surprising. Just remove any one bridge (as in Figure 2.5)

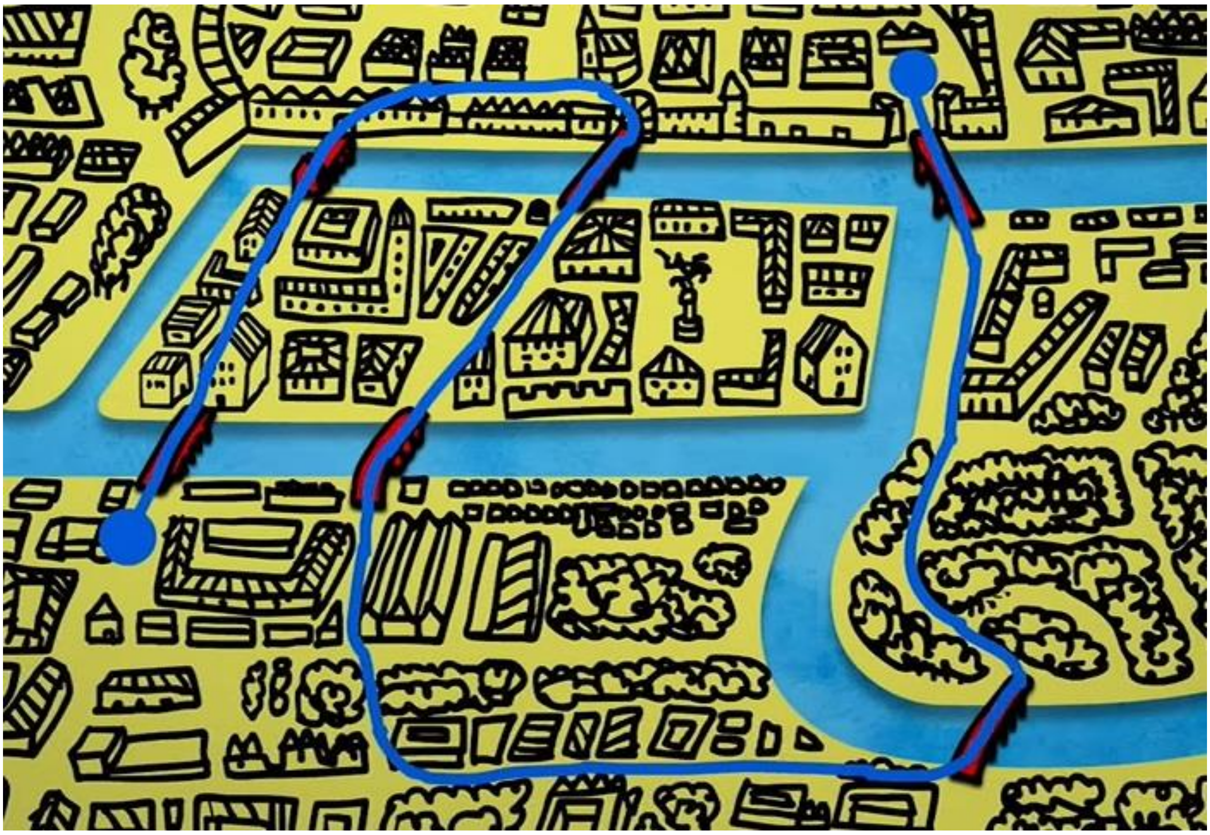

Figure 2.5. ***How to create a Euler Path in Königsberg***

Fun facts: History made a Euler path of its own. During World War II, the Soviet Air Force destroyed two of the city's bridges so that a Euler path was created easily. These bombings significantly wiped Königsberg off the map, and it was later reconstructed as the Russian city of Kaliningrad. Although Königsberg and seven bridges may not exist anymore, they will be remembered as a part of history through the breakthrough which led to the emergence of a whole new theory in Mathematics.

C. Eulerian Graph and Semi-Eulerian Graph

- **Definition 2.3:** *Eulerian Graph* is a connected graph having a *Euler Circuit* which includes every edge of the graph. (Examples in Figure 2.6 and Figure 2.7)
- **Theorem 3:** A graph is a *Eulerian Graph* if and only if each vertex has an even degree.
- **Definition 2.4:** *Semi-Eulerian Graph* is a connected graph containing a Euler path but an Euler circuit and this Euler path includes every edge of the graph. (Examples in Figure 2.8)
- **Theorem 4:** A graph is a *Semi-Eulerian Graph* if and only if there is one pair of vertices with odd degree.
- **Theorem 3:** and Theorem 4 are both based on the theorem 1 and theorem 2 of Euler Path and Euler Circuit.

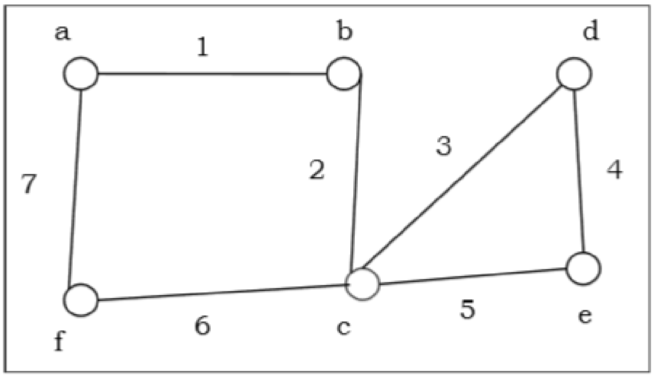

Figure 2.6. ***Eulerian Graph***

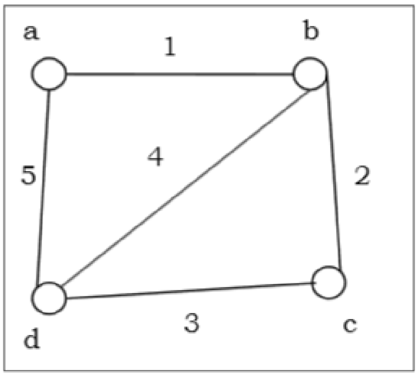

Figure 2.7. ***Non-Eulerian Graph***

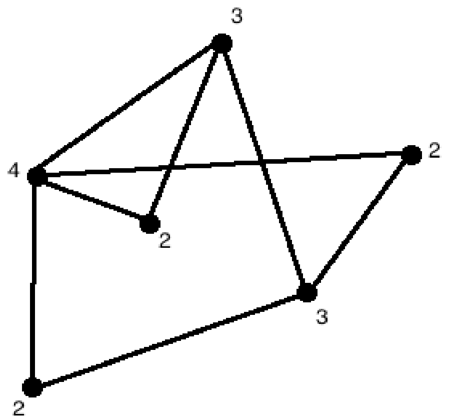

Figure 2.8. ***Semi-Eulerian Graph***

More puzzles on graph theory

A. Five-room puzzle

1. Problem

This popular puzzle involves a large rectangle divided into five "rooms". The objective is to cross each "wall" of the diagram with a continuous line only once. Let's take a look at Figure 2.9 and Figure 2.10.

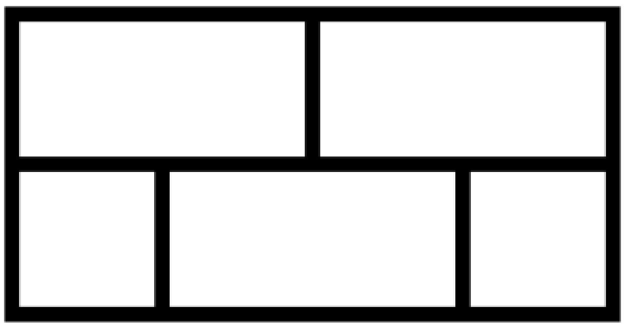

Figure 2.9. ***A simple rendition of Five-room puzzle***

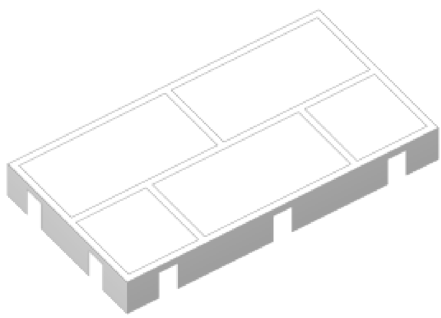

Figure 2.10. ***3D rendition of the rooms and doors***

2. Solution

- How can we solve this puzzle? As with the Seven Bridges of Königsberg, the puzzle is represented in graphical form with each room corresponding to a vertex (including the outside area as a room) and two vertices joined by an edge if the rooms have a common wall. Figure 2.11 has demonstrated the comparison between the graphs of the Seven bridges of Konigsberg (left) and Five-room puzzles (right). The numbers denote the degree of each vertex. Vertices with an odd degree are shaded orange.

- Since there are more than two vertices with an odd number of degrees, the resulting graph does not contain an Euler Path nor an Euler Circuit (Based on theorem 1 and theorem 2), which means that the solution for this puzzle is impossible.

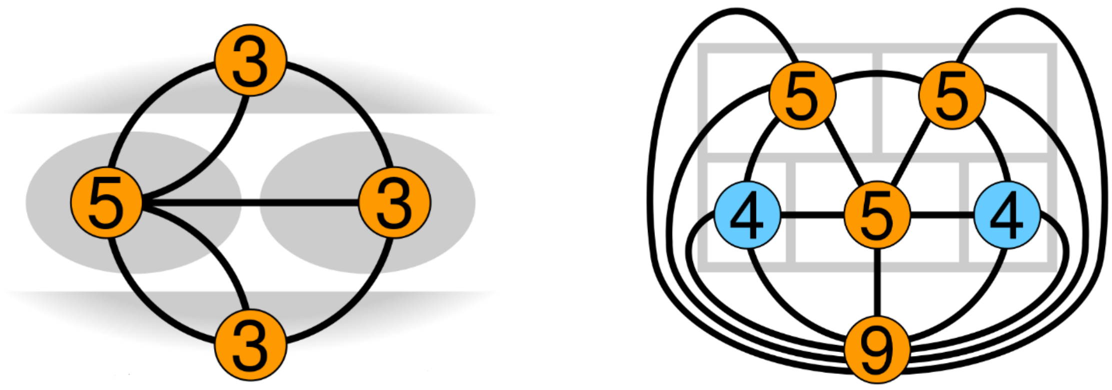

Figure 2.11. ***Graph of the seven bridges of Konigsberg and Five-room puzzle***

Applications

1. Eulerian Circuit recovers a genome sequence

- As you will see the application of Hamiltonian Path in the next chapter, it can identify the overlapping DNA sequence from its fragments and reconstruct the original genome. Eulerian Circuit is considered as the more efficient approach than the later one.

- Assume we have a million nodes in the network, this means that we are going to have an enormous network of a million reads. Even though Hamiltonian Circuit seems very similar to Eulerian Circuit, it will take so much time to solve the network in this way. We can form a network as follows:

- Create nodes for each ***distinct*** prefix/suffix from reads. (Figure 2.12 and 2.13)

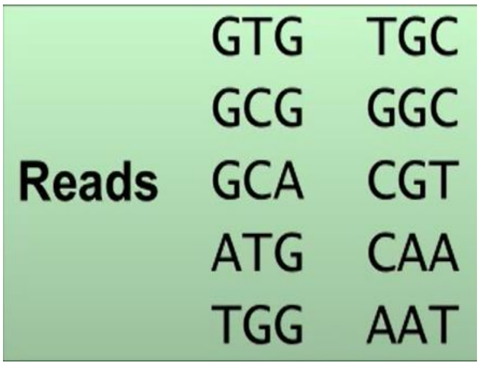

Figure 2.12. ***DNA reads***

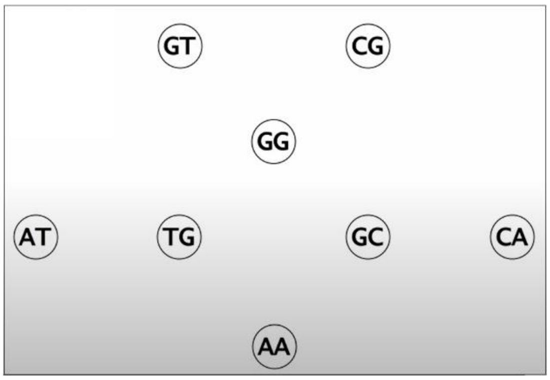

Figure 2.13. ***Distinct prefix/suffix diagram***

- Connect node w to node v with a directed edge if there is a read whose prefix is w and whose suffix is v. (Figure 2.14)

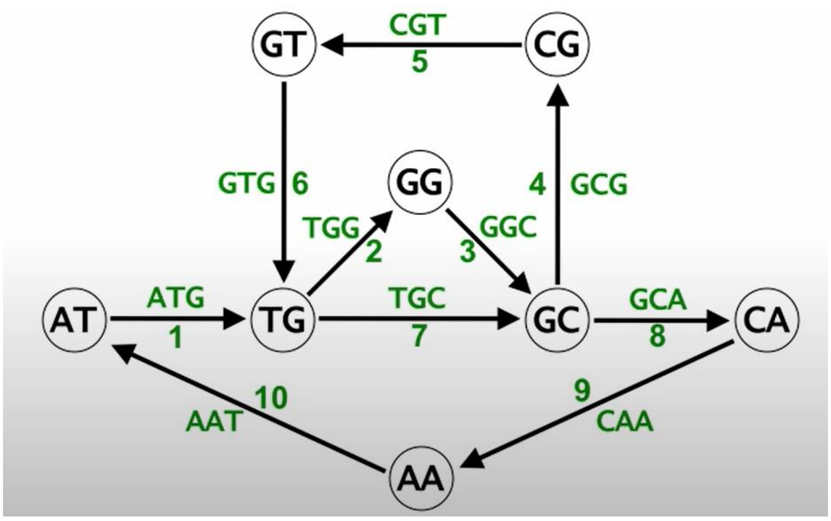

Figure 2.14. ***Connected nodes graph***

- We have an Eulerian Circuit in this network (Figure 2.14). Based on that, we can find the sequence of reads below:

  ATG→TGG→GGC→GCG→CGT→GTG→TGC→GCA→CAA→AAT

- Therefore, the original Genome is ATGGCGTGCA.

- The key thing is that computers can find Eulerian Circuit very quickly even if the network is very big. This method serves as the foundation for sequencing algorithms that are running around the world and genomes from all various kinds of species as well as humans.

- A genome sequence does contain some clues about where genes are, even though scientists are just learning to interpret these clues. Scientists also hope that being able to study the entire genome sequence will help them understand how the genome as a whole works—how genes work together to direct the growth, development, and maintenance of an entire organism.

2. Eulerian graph is also applied in designing the floor.
3. There are some algorithms for processing trees that rely on an Euler tour of the tree (where each edge is treated as a pair of arcs)

HAMILTONIAN GRAPHS:

History

Hamiltonian graphs, cycles and paths were named after William Rowan Hamilton who contributed to the fields of pure mathematics and mathematics for physics. He also invented the Icosian Calculus in 1857, which is used to find Hamiltonian Paths on a dodecahedron.

Figure 3.1

The Hamiltonian Cycle Problem is in fact a special case of the Travelling Salesman Problem. The Travelling Salesman Problem asks the question: 'In a list of *n* cities with the distances between each pair, what is the shortest possible route that visits each city exactly once and returns to the origin city?'
By setting the distance between each city to one unit, if the total distance is *n* a Hamiltonian circuit shall exist.

Even earlier, Hamiltonian cycles and Knight's Graphs had been studied in the 9th century in Indian Mathematics by Rudrata (a Kashmiri poet and literary theorist) and Islamic Mathematics by al-Adli ar-Rumi (an Arab player and theoretician of an old form of chess known as *Shatranj*)

Theory

If there exists a closed walk in the connected graph that visits every vertex of the graph exactly once (except starting vertex) without repeating the edges, then such a graph is called as a Hamiltonian graph.

Here is a simple example of a Hamiltonian Graph:

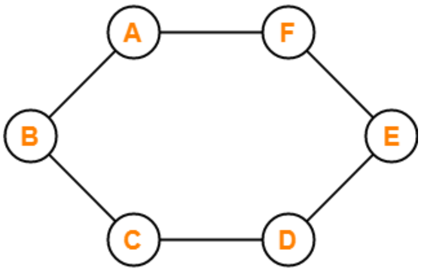

***Figure* 3.2**

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow A$

is the closed walk contained in the graph

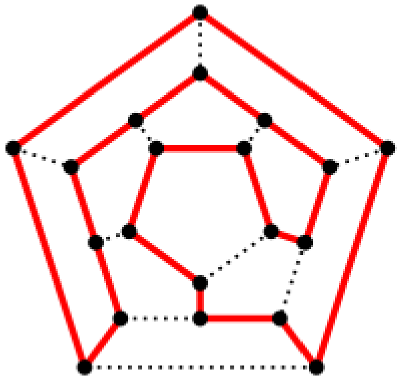

Figure 3.3

Although some edges have not been travelled,

this graph is still Hamiltonian as all vertices have been visited

Hamiltonian Path:

If there exists a walk in the graph that visits every vertex of the graph exactly once without repeating edges, it is known as a Hamiltonian path.
(It is to be noted that the path is not a closed walk)

Hamiltonian Circuit/Cycle:

If there exists a walk in the connected graph that visits every vertex of the graph exactly once (except starting vertex) without repeating the edges and returns to the starting vertex, then such a walk is called as a Hamiltonian circuit.

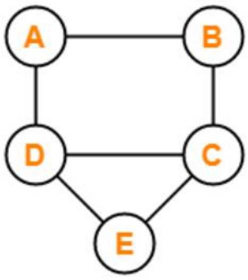

Figure 3.4

Hamiltonian Path: $A \to B \to C \to D \to E$
Hamiltonian Cycle: $A \to B \to C \to D \to E \to A$

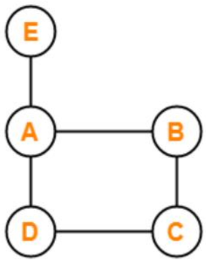

Figure 3.4

Hamiltonian Path: $E \to A \to B \to C \to D$
Hamiltonian Cycle does not exist

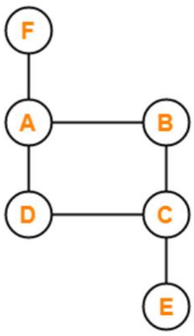

Figure 3.5
Hamiltonian Path does not exist
Hamiltonian Cycle does not exist

Knight's Tour

History:

The earliest known occurrence of the Knight's Tour sequence dates back to 9th century AD in Rudrata's *Kavyalankara*. The Sanskrit work uses Indic writing where each syllable represents the square of a chessboard. We will further discuss verse-pair tours composed as Knight's Tours by Vedāntadeśika and Rājānaka Ratnākara.

Leonhard Euler was one of the first mathematicians to investigate the Knight's Tour Problems. The first path for completing the Knight's Tour was given by H. C. von WarnsDorff in 1823.

Theory:

The knight on a chessboard moves in the following ways:

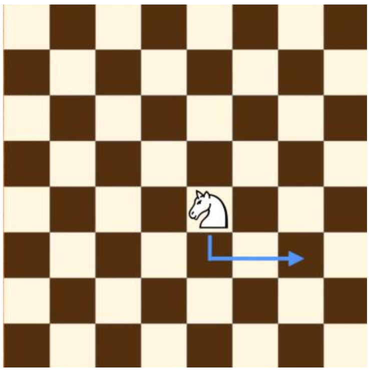

Figure 3.7

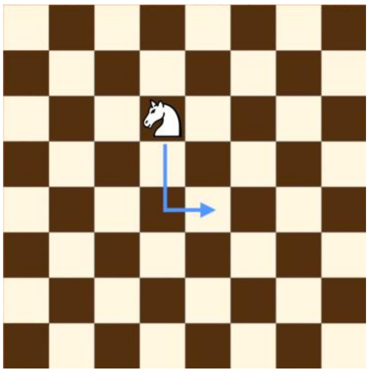

Figure 3.8

Since it can move in any direction, the possible paths are shown as below:

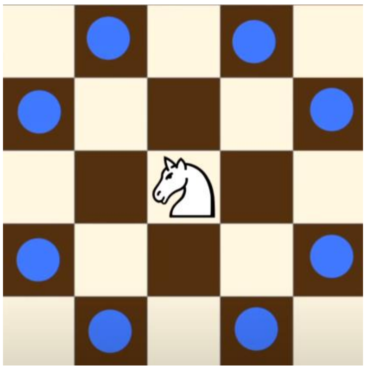

Figure 3.9

The Knight's Tour involves moving the knight around the chessboard such that it visits every square exactly once (resembles a Hamiltonian Path).

If the knight ends on a square it began with, the tour is closed and can be repeated. Else the tour is open.

Closed Tour

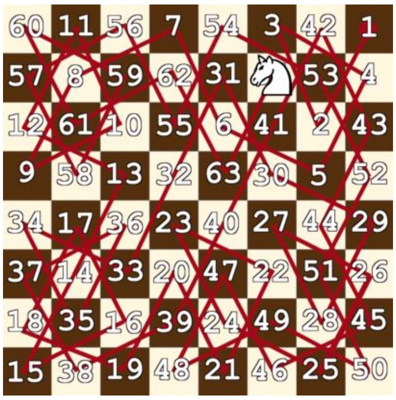

Figure 3.10

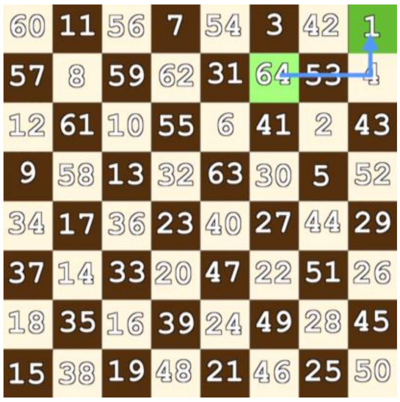

Figure 3.11

Open Tour

Figure 3.12

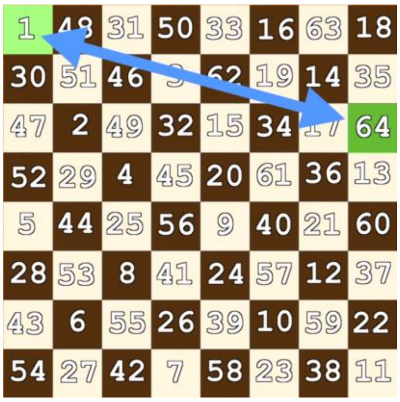

Figure 3.13

An interesting pattern that was found was that an open tour displayed the properties of a semi-magic square.

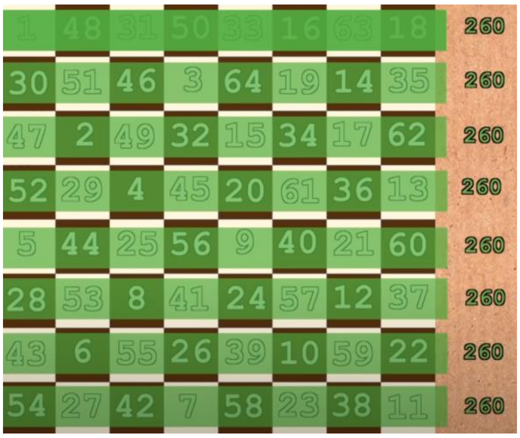

Figure 3.14

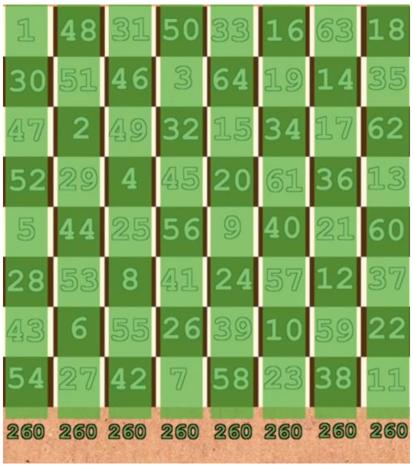

Figure 3.15

The rows and columns sum up to 260**.**

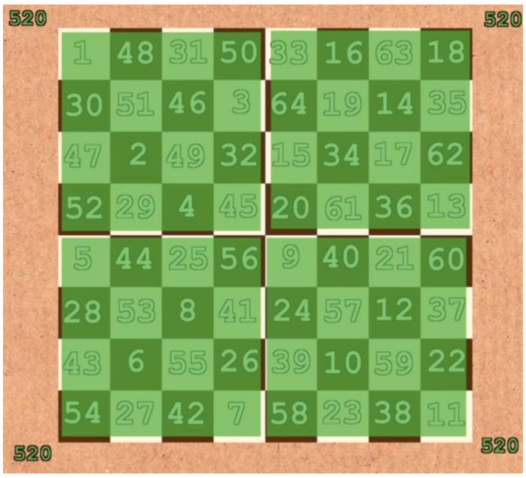

Figure 3.16

Quadrants sum up to 520

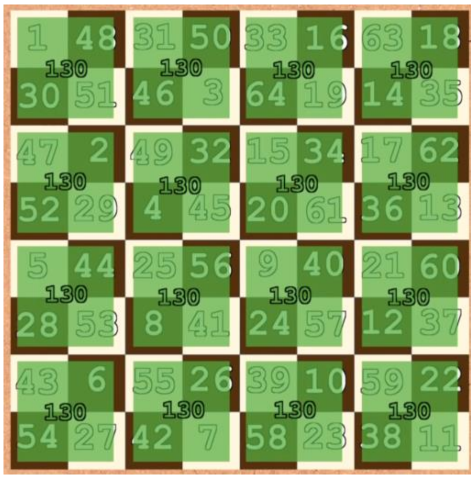

Figure 3.17

Each of the 2 x 2 squares sum up to 130

Methods of Finding Tours

On an 8×8 board, there are 26,534,728,821,064 known directed closed tours (i.e. two tours along the same path that travel in opposite directions are counted individually). The number of undirected tours are halved since the path of each tour is reversed.

Warnsdorff's Rule:

Under this rule, the knight will always move to a square whose degrees are the least (i.e., the number of paths forward are the least).

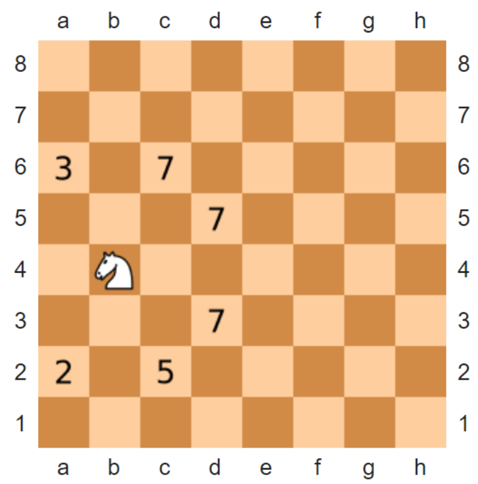

Figure 3.18

In the case above, the knight will move from 4b to 2a as it is the vertex with the least degree. Such an algorithm can be designed in the form of a computer program as well.

Verse Pair Tours

Vedāntadeśika who lived in the second half of the 13th century was a gifted poet among the Sanskrit Poets. His literary work Pādukāsahasram was a hymn to the Sandals of Shriram which were worshipped by Bharata. In his work, a pair of versed use the pattern of Knight's Tour as shown below:

स्थिरागसां सदाराध्या विहताकततामता
सत्पादुके सरासा मा रङ्गराजपदं नय ॥ → *sthirāgasāṃ sadārādhyā vihatākatatāmatā* /
satpāduke sarā sāmā raṃgarājapadannaya //

Which translates to:

[Oh Sandal-pair of the Creator! You are to be worshipped all the time by the ordinary folk whose sins are ever present. You demolish the sorrows and unpleasant happenings. Lead me jingling to the feet of God]

Each syllable can be ordered along the checks of the chessboard as shown below. Using the properties of the Knight's Tour a new verse can be generated.

स्थि1	रा30	ग9	सां20	स3	दा24	रा11	ध्या26
वि16	ह19	ता2	क29	त10	ता27	म4	ता23
स31	त्पा8	दु17	के14	स21	रा6	सा25	मा12
रं18	ग15	रा32	ज7	प28	द13	न्न22	य5

Figure 3.19

Following the path from 1 → 32 we obtain the following verse:

'स्थिता समयराजत्पा गतरा मादके गवि ।
दुरंहसां सन्नतादा साध्यातापकरासरा ॥ → *sthitā samayarājatpā gatarā mādake gavi* /
duraṃhasāṃ sannatā dā sādhyātāpakarāsarā //

Which translates to:

[You protect those who shine in their timely conduct. You are stationed in the Sun. You receive riches. You remove the grief of hard sinners. Your devotees feel intoxicated by your Grace. Your rays render lack of distress possible. You move all around.]

Similarly, it has been noticed that poets such as Rudrata and Bhoja use a common approach of the Kinght’s Tour as Vedāntadeśika does. The poet Ratnakara also employs the Knight’s Tour in his verses however uses them differently.
It is impressive to see the solution of Knight’s Tour being used in Literary Arts as early as the 9th Century. The patterns from Mathematics are infact extensively used in the fields of Arts and Music.

Applications

II. A very simple application is mapping a bus route. For example, if a school bus picks up *x* students on the way to school, the most efficient way to do so is by identifying a Hamiltonian path such that each student’s stop is visited only once and none of the paths are repeated.

III. Since the Hamiltonian Path guarantees the shortest path to visit all vertices, it is widely used in the GPS to track the shortest and optimized path to the user’s destination.

IV. Genome Mapping uses Hamiltonian maps extensively. Given below is an example of the same:
For example, we are given random DNA reads as shown. Using these we can identify the overlapping sequence and construct the original genome. Using the Hamiltonian network, we can do so. Each DNA read can be represented as a network.

V.

Figure 3.20

We identify the overlapping DNA fragments as the various nodes. For example, if ATG is the prefix then its can be joined with TGG and TGC. Using this concept, we plot the path:

$$ATG \rightarrow TGG \rightarrow GGC \rightarrow GCG \rightarrow CGT \rightarrow GTG \rightarrow TGC \rightarrow GCA \rightarrow CAA \rightarrow AAT \rightarrow ATG$$

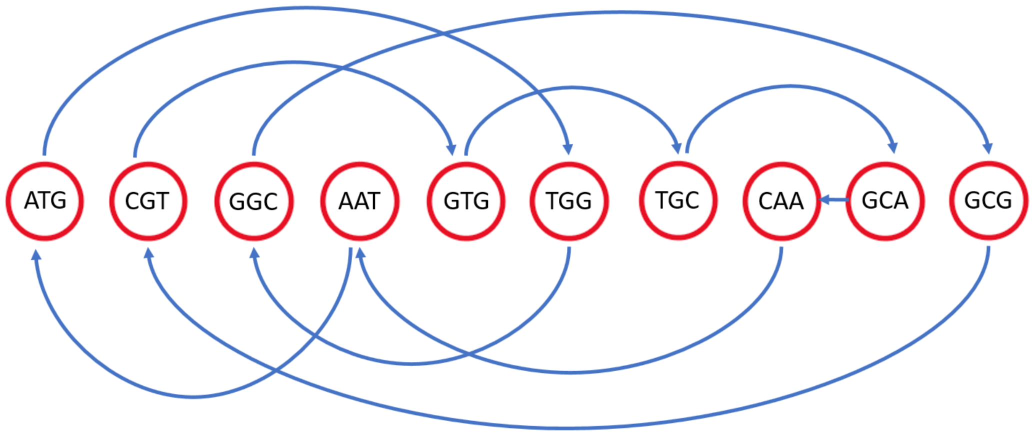

Figure 3.21

We can represent this as:

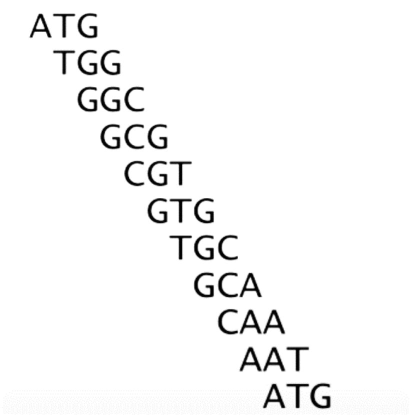

Figure 3.22

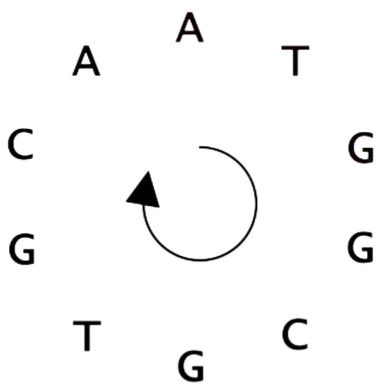

Figure 3.23

Hence the original Genome obtained linearly is: ATGGCGTGCA
However, the Euler’s Graph is more effective when used.

VI. It is also used in computer graphics and designing electronic circuits.

DIJKSTRA'S ALGORITHM

Background Behind Dijkstra's Algorithm

The Dijkstra's Algorithm was found and presented by Edsger Dijkstra in 1956 for the inauguration of the ARMAC computer and was later published in 1959 as "A Note on Two Problems in Connexion with Graphs" in the Numerische Mathematik, edited by F.L. Bauer. Dijkstra was inspired to solve the shortest path problem when he and his fiancée were tired after shopping.

He asked the question, "What's the shortest way to travel from Rotterdam to Groningen?" Dijkstra designed the algorithm without pencil and paper in 20 minutes. In his paper, Dijkstra aimed to solve two problems: constructing the tree of minimum total length between the n nodes, and finding the path of minimum total length between two given nodes P and Q. The first problem's solution was based on an algorithm initially found and developed by a Czech mathematician named Vojtěch Jarník.

This algorithm was later republished by Robert C. Prim in 1957 and Edsger Dijkstra as a solution to the first problem in "A Note on Two Problems in Connexion with Graphs". This algorithm generated a minimum spanning tree, which is a tree of edges between each vertex where the total weight is minimized. The second problem's solution was an algorithm that Dijkstra designed himself and was later used for the foundation of many path-finding applications.

Edsger Dijkstra

Edsger Wybe Dijkstra was a Dutch computer scientist who was mostly known for his work on the Dijkstra's shortest-path algorithm. He was born in Rotterdam, Neth on May 11, 1930 to a mother and father who were a mathematician and a chemist, respectively. He pursued an undergraduate education in the University of Leyden and completed his degree in mathematics and physics at age 21. After a recommendation by his father, he joined a summer school program for programming electronic computing devices at Cambridge, given by Maurice Wilkes who was instrumental in Cambridge's EDSAC, which was the first full-size stored-program computer.

The summer program he attended made Dijkstra the world's first Dutch computer programmer. Following his short venture into computer programming, he began working at the Mathematisch Centrum (now known as the Centrum Wiskunde & Informatica, or National Research Institute for Mathematics and Computer Science), a research center in Amsterdam. During his 10 years in the Mathematisch Centrum, he built early computers

such as the Automatic Relay Calculator Amsterdam (ARRA), the successor of the ARRA known as the ARRA II, a variation of the ARRA II known as the Fokker's First Calculator Type ARRA (FERTA using Dutch initials), and the Automatic Computing machine of the Mathematica Center (ARMAC using Dutch initials)—the last of which was a computer that was important to Dijkstra’s contribution to computer science. During the ARMAC computer’s inauguration in 1956, Dijkstra showed off an algorithm that he formulated to solve the Shortest Path problem, which was later published in 1959 due to the lack of automatic computing journals at the time.
Dijkstra also constructed an efficient algorithm for the Minimum Spanning Tree. Both algorithms were published and was titled a Citation Classic in the Web of Science database in the period 1945-2000.

Dijkstra also worked on the Ph.D. thesis that would come to be the foundations of asynchronous computing and by extension, modern operating systems. This was an important discussion, as computer manufacturers at the time faced the problem of the Central Processing Unit and other computer parts computing asynchronously. His Ph.D. thesis brought about and considered multiple techniques that would buffer communication between peripheral devices that operated at greatly differing speeds.

Still at the Mathematisch Centrum, Dijkstra and a colleague named J. A. Zonneveld created a compiler for Algol-60, an important programming language at the time. Algol-60 was a landmark publication in computer science as it brought around new innovations to the field, particularly a nested block structure and a mathematically exact notation called the Backus-Nair form.
Later in the 60s, Dijkstra left the Mathematisch Centrum to become a professor of mathematics in the Eindhoven University of Technology. During his tenure at the EUT, he created the THE (Technische Hogeschool te Eindhoven) operating system, which has impacted all the subsequent operating systems through influences in design and operation.

Dijkstra's Algorithm

Dijkstra's Algorithm considers n points, otherwise known as nodes, with some or all pairs of points being connected by a branch that has a given length or weight. Every point must have a path, meaning that no point can be disconnected or disjoint. There must be a source node P and an end node Q. The algorithm works on both directed and undirected graphs.

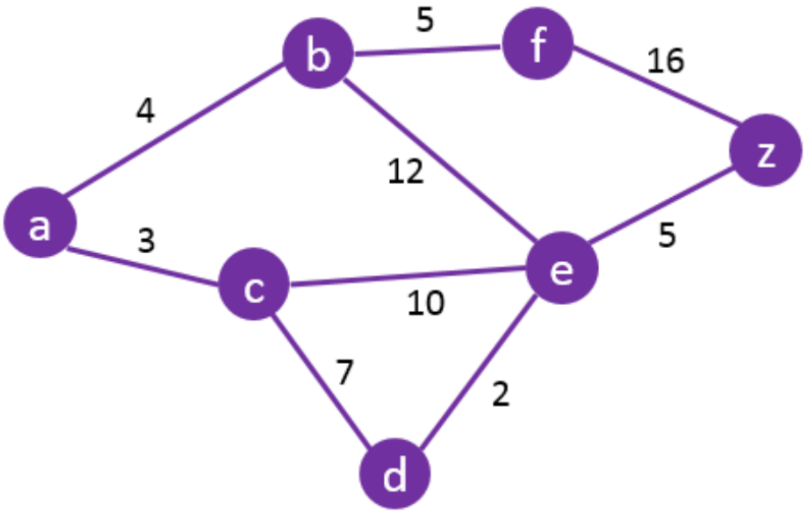

Figure 4.1 An undirected graph with given lengths

The algorithm uses six different sets to categorize nodes and branches that are parts of the possible paths to the end node Q. These sets are:

1. **Set A** — the nodes of which the minimum length path from source node P is known; nodes will be added to this set-in order of increasing minimum path length from node P
2. **Set B** — the nodes of which are connected to nodes in set A; these nodes are the next nodes to be added to set A
3. **Set C** — the remaining nodes
4. **Set I** — the branches occurring in the minimal paths from node P to the nodes in set A
5. **Set II** — the next branches to be placed in set I; one and only one branch of this set will lead to each node in set B
6. **Set III** — the remaining branches (rejected or not considered)

Initially, all nodes start in set C and all branches start in set III. The source node P must then be transferred to set A to begin.

Step 1. Check all the branches r connecting the node just transferred to set A with all the nodes R in sets B or C. If node R is in set B, check whether using branch r gives a shorter path than the known path that uses the corresponding branch in set II. If it is longer, then branch r is rejected; if it is shorter, then it replaces the corresponding branch in set II, and the previous corresponding branch is rejected. If node R belongs to set C, it is added to set B and branch r is added to set II.

Step 2. Using only branches from set I and one from set II, every node in set B can be connected to node P in only one way. The node in set B with minimum distance from P is transferred to set A, and the corresponding branch is transferred from set II to set I. The process is then repeated until end node Q is finally transferred to set A. Only then will the solution be found.

The algorithm was initially reported to have a time complexity of $O(V^2)$ with V as the number of points or vertices, however upon closer inspection and observation, with binary heap for priority queue implementation it has a time complexity of O(ELogV) with E as the number of branches or edges. If the algorithm is implemented with Fibonacci heap, the time complexity reduces to O (E + VLogV).

Other Algorithms in Comparison

There are multiple other algorithms that attempt to solve the shortest path problem. There is no definite best algorithm as each has its own advantages and disadvantages and are the most useful in only specific situations.

The Bellman-Ford Algorithm

The Bellman-Ford algorithm was initially proposed in 1955, around the time Dijkstra's algorithm was proposed as well. Unlike Dijkstra's algorithm, the Bellman-Ford algorithm can handle edges with negative weights. However, the algorithm stores data for branches not included in the shortest path, which are the branches in set I and II. Because of this, the Bellman-Ford is slower with a time complexity of $O(|V|\cdot|E|)$.

Floyd-Warshall Algorithm

Unlike the two algorithms previously discussed, the Floyd-Warshall algorithm is not a single-source algorithm. Instead of calculating the shortest distance from each node to the source node, the algorithm calculates the minimum distance between every combination of pairs of nodes in the graph. This is useful for multi-stop paths as it already calculates the shortest distance for the nodes that are not the first source node. However, it does have a time complexity of $O(V^3)$ which is much higher than the Dijkstra's algorithm's $O(|E|\cdot Log(|V|))$.

Johnson’s Algorithm

Johnson’s algorithm is quite similar to Floyd and Warshall’s algorithm, in which it also calculates the shortest distance between every pair of nodes. This algorithm actually uses two other previously discussed algorithms as subroutines. It uses the Bellman-Ford algorithm to reweight the graph in order to eliminate negative edges and negative cycles. It then uses Dijkstra's algorithm to calculate the shortest distance between all pairs of points. The time complexity is ($V^2 \cdot \log(V)+|V|\cdot|E|$), which means that if the number of edges is small, then it runs faster than the Floyd-Warshall Algorithm. This edge dependency makes Johnson's algorithm much better for sparse graphs and worse for dense graphs.

Dijkstra’s Algorithm’s Real-World Influence

The Dijkstra’s algorithm and other shortest path algorithms have had an influence and impact within many important facets of the world, specifically its economy. Many companies, such as Amazon, Lazada, Flipkart, Shopee, Alibaba, and many more use Dijkstra's algorithms or a combination of multiple algorithms to calculate the most efficient path of delivery. The courier, express, and parcel (CEP) market size worldwide was 330.34 billion euros in 2019 and continues to grow even more fervently with the recent COVID-19 crisis.

With the stay at home and quarantine orders worldwide, the shift to e-commerce from physical stores has been reported to have accelerated 5 years, which provides even more business for the CEP and logistics market. With the shortest path algorithms, many multibillion-dollar companies are saving both time and money.

Because of its foundational stature, many applications such as Waze and Google Maps use a modified form of Dijkstra’s Algorithm. Waze and Google use proprietary algorithms that have more than weighted edges. Their algorithms use a combination of past and real-time data that contain factors such as live traffic, major traffic events, the time of day, and many other variables.

Now, better algorithms are being developed using new technology such as artificial intelligence.

CONCLUSION

There are many ways Graph Theory is used in our daily lives. Data structures are largely dependent on graph structures and algorithms. A graph can be assigned the dimension of weights (such graphs are known as weighted graphs) and are used to program GPS, travel planning search engines and flight times/costs. Many problems in Graph Theory also use various colouring graphs such as – Four-colour theorem, Total colouring conjecture, Strong perfect graph theorem, etc. Route problems are solved using Hamiltonian graphs and Euler's Cycles. Many applications also include Network Flow, Visibility Problem (Museum Graph problem using the Chvátal's art gallery theorem), Decomposition Problems (breaking down a graph into subgraphs eg. Arboricity where minimum number of forests must be partitioned from edges).

Graph Theory has helped us derive numerous efficient algorithms and visualize/represent a problem effectively. Although we merely touched the surface with our research, it further encouraged us to dive into the depths of Graph Theory. The few theorems and puzzles mentioned above gives us a better understanding of the applications of graph theory which ranges from simple problems on a large scale such as mapping bus routes to detailed and dynamic problems such as genome mapping. Graph Theory is extensively used in every Mathematical and Scientific field and is yet to solve many unsolved problems.

REFERENCES:

Euler Graph and Circuit:

- https://en.wikipedia.org/wiki/Eulerian_path
- https://www.codespeedy.com/eulerian-path-and-circuit-for-undirected-graphs-in-cpp/
- https://www.youtube.com/watch?v=dSK5jTEe-AM&list=PLHXZ9OQGMqxersk8fUxiUMSIx0DBqsKZS&index=81
- https://www.youtube.com/watch?v=WY6Jsp3NxIk&t=1238s
- https://www.youtube.com/watch?v=5M-m62qTR-s&t=156s
- http://www.csc.villanova.edu/~japaridz/Archive/1300/lect7.5/tsld010.htm
- http://discrete.openmathbooks.org/dmoi2/sec_paths.html
- https://www.tutorialspoint.com/eulerian-graphs
- http://mathonline.wikidot.com/eulerian-graphs-and-semi-eulerian-graphs#:~:text=Definition%3A%20A%20graph%20is%20considered,return%20to%20the%20start%20vertex.
- https://en.wikipedia.org/wiki/Five_room_puzzle
- https://www.youtube.com/watch?v=0JlUy_l-RTk
- https://www.ijarse.com/images/fullpdf/1486979678_G1035ijarse.pdf
- http://www.genomenewsnetwork.org/resources/whats_a_genome/Chp2_1.shtml

Hamiltonian Graphs:

- https://mathworld.wolfram.com/news/2003-08-06/magictours/
- https://www.johndcook.com/blog/2011/04/06/a-knights-magic-square/
- https://core.ac.uk/download/pdf/81964499.pdf
- https://en.wikipedia.org/wiki/Graph_theory#:~:text=In%20mathematics%2C%20graph%20theory%20is,also%20called%20links%20or%20lines).
- https://en.wikipedia.org/wiki/William_Rowan_Hamilton
- https://www.ias.ac.in/article/fulltext/reso/025/08/1095-1116
- http://www.mayhematics.com/t/t.htm
- https://www.youtube.com/watch?v=ab_dY3dZFHM

Dijkstra's Algorithm:

- https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/
- https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/
- http://www.computinghistory.org.uk/det/4179/Edsger-Dijkstra/
- https://www.sciencedirect.com/topics/computer-science/dijkstra-algorithms
- http://www-m3.ma.tum.de/foswiki/pub/MN0506/WebHome/dijkstra.pdf
- https://rosettacode.org/wiki/Dijkstra%27s_algorithm
- https://medium.com/@farruk/practical-dijkstras-algorithm-b329ade79a1e
- https://isaaccomputerscience.org/concepts/dsa_search_dijkstra
- https://www.theguardian.com/news/2002/aug/19/guardianobituaries.obituaries1
- http://www.csl.mtu.edu/cs2321/www/newLectures/30_More_Dijkstra.html

Others:

- https://en.wikipedia.org/wiki/Graph_theory
- Essentials of Discrete Mathematics by David Hunter