Applications of Graph Theory

I first came across the mathematical concept of Graph Theory a while ago whilst listening to 'A Brief History of Mathematics', a BBC 4 podcast by mathematician Marcus du Sautoy. In one of the earlier episodes of the podcast, he describes the story of Leonhard Euler's solution to a popular 18th century conundrum, The Seven Bridges of Königsburg (Figure 1). Euler's approach to solving this problem ultimately led to the origin of graph theory, which was only one very small part of his overall contribution to the discipline of mathematics during his lifetime. Graph theory is now seen in many real-world applications such as the London underground, circuit design and navigation systems such as Google Maps.

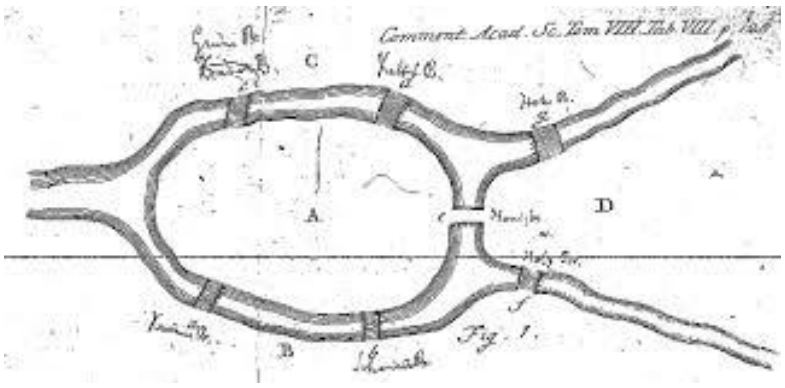

Figure 1: Euler's original drawing

Figure 1

For those not familiar with this story, in the 18th century, the Prussian city of Königsburg was split into four separate land masses, connected by seven bridges. The problem was to find a way to walk across all the bridges only once and get back to where you had started (this is now known as an eulerian circuit in graph theory).

In 1736 Leonard Euler went about solving this problem. He recognised that it could not be solved through using the usual concepts of mathematics such as geometry, algebra or even the art of counting, and so set about approaching it in a whole new manner entirely. He chose not to go with the common approach of attempting to take every possible route until he found one that worked, but instead visualised the problem as a network, where the bridges are lettered from a to g and the land masses from A to D. This approach of looking at the problem not in terms of measurement, but in terms of geometry of position was something that prominent mathematician, Gottfried Leibniz, had previously sought to establish. Euler represented the separate land masses as points (vertices) and the connecting bridges as lines (edges) joining one land mass to another. He then looked at the number of edges incident on a given vertex, or the degree of the vertex as it is now referred to as. If we take land mass D, for example, you find that there are three bridges, or edges, joined to it. This means that it has degree three, or an odd degree. If you count the edges incident on each other land mass, you will find that they also all have an odd number of bridges coming from them, and so there are four odd vertices. This then formed Euler's proof that such a path could not exist, as you would need to begin or end the trip at any odd vertices. An odd vertex must be the start and end point of an eulerian circuit, as it takes one edge to travel to a vertex and one to leave it, and so a vertex with an odd degree cannot be passed through without leaving an untraversed edge, unless you start or finish with that vertex. With there only being one beginning and one end, you can therefore only include two odd vertices at most in a trip, and only one for an eulerian circuit. Having four odd vertices made this impossible to do without going over a bridge twice (repeating an edge or arc), and so the conclusion was made that it could not be done.

This solution then led to the origin of the area of discrete mathematics known nowadays as graph theory. This branch of mathematics looks at networks of points connected by lines, otherwise known as graphs, and their relations. In graph theory, a graph is defined as a set of vertices (points or nodes) and edges (lines) that can connect the vertices.

A graph,G, is a pair $G = (V, E)$, where $V$is the set with vertices as its elements, and E is the set with edges (or paired vertices) as its elements.

For instance, the graph $G = (V, E)$ with the set $V: \{v_1, v_2, v_3\}$ and set $E: \{v_1v_2, v_2v_3, v_3v_1\}$ would look like the graph shown in Figure 2, which is also known as the complete graph called K_3. The pairs of vertices in the edge set represent the end-points of a given edge. A complete graph is a graph in which every vertex is directly connected by a single edge to each of the other vertices, and K_n is the complete graph with n vertices.

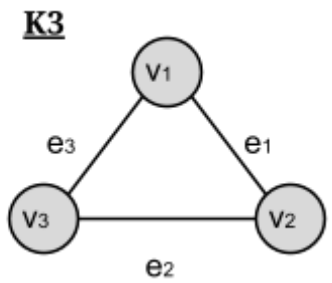

Figure 2: K3

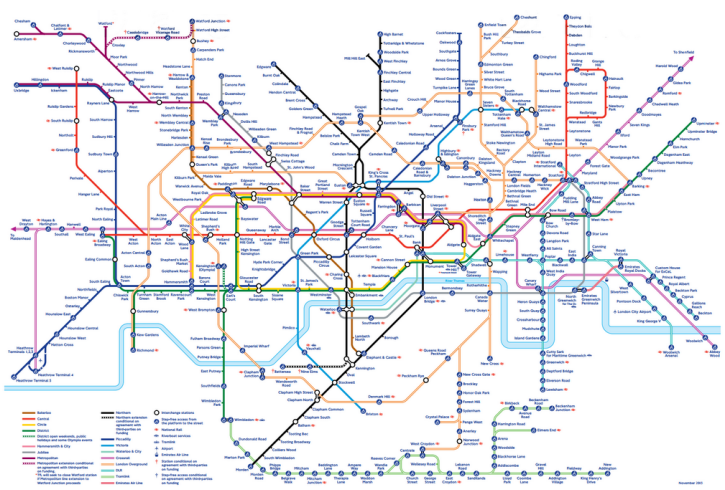
Figure 3: London Underground Tube Map

The London underground network map is a good illustration of basic graph theory being applied in everyday life. It is an example of a topological map, which is a type of diagram that has been simplified so that it lacks scale, and the distances and directions can be modified, but it still maintains the relations between points. In the tube map, the stations are the vertices of the graph, and the connecting railways are the edges. Interchanges are also shown through where multiple edges are incident on a given vertex. This way of representing the underground network adds simplicity and clarity through the distortion of distance and direction, so it focuses on traversing between different vertices rather than the geographical accuracy of the map.

There are multiple different types of graph, including directed graphs (digraphs) and undirected graphs, simple graphs or planar graphs and so on, which all have their own different properties and applications. Planar graphs are particularly useful when looking at designing circuits, subways or utility lines for example, where edges crossing at points other than connected vertices are a nuisance, as they are graphs that can be drawn without any edges crossing. A a good example of this is the complete graph K_4(Figure 4), which is usually seen drawn with some of its edges intersecting at points other than at vertices. However, it can in fact be redrawn as a planar graph, known as its planar representation. To determine whether or not a graph is planar we can use Euler's theorem for planar graphs:

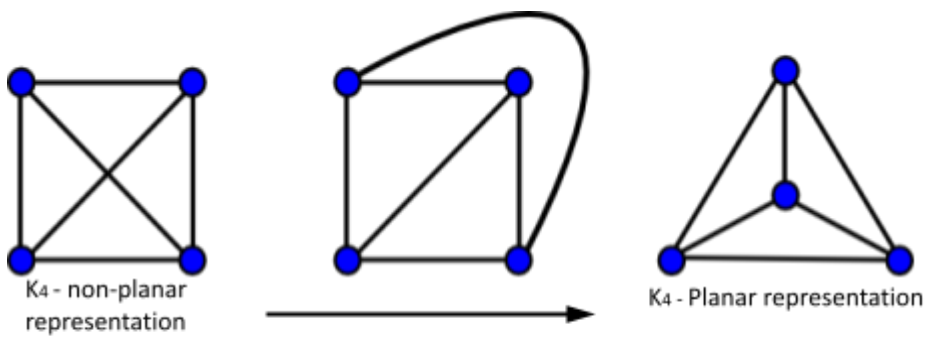

Figure 4: K4

Let $G =< V, E >$ be a connected planar graph with $|V| = v$(number of vertices) and $|E| = e$ (number of edges). Let the letter$r$be the number of regions determined by the planar embedding.

Then we can conclude that:

$v - e + r = 2$

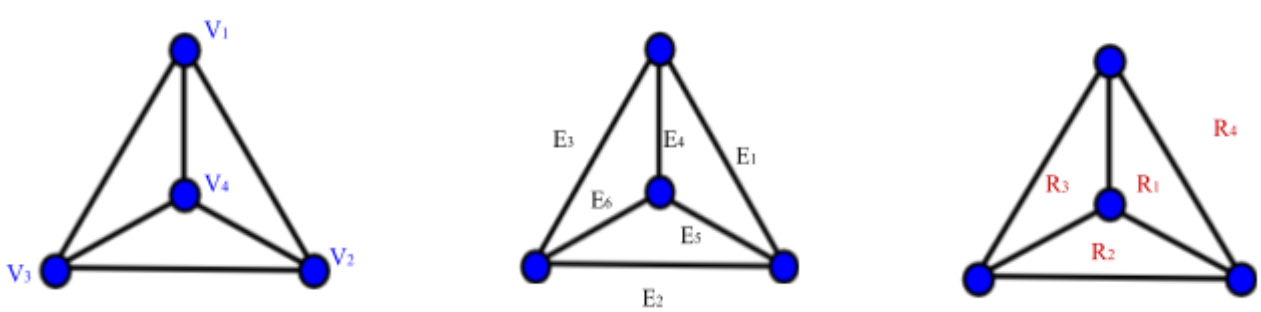

Figure 5: Vertices, edges and regions of K4

This theorem is proved by method of induction, but if a we use K_4 as an example of a known planar graph (figure 5), the number of vertices is 4, the number of edges is 6, and there are 4 regions (including the infinite region) when the graph is drawn as a planar representation. Substituting these values in $4 - 6 + 4 = 2$, so we see the equation holds true for K_4. If the graph is not drawn as a planar representation, the number of regions may be altered by edges crossing at points other than vertices.

Graphs can also have algorithms applied to them for useful applications such as to find a shortest path through a weighted graph along its edges from one chosen vertex to another. Algorithms like these are an integral part of applications such as Google maps and SAT NAV systems. Taking Google Maps as an example, road networks can be modeled as huge graphs, with the roads as edges and vertices being the points where these roads intersect. Various algorithms can be applied to these graphs to find the most direct route for example. In this instance giving the edges of a graph values (weights or costs), such as distance or an estimated time to travel along an edge from one vertex to another, can allow for shortest routes to be calculated. One such algorithm that is used for this is Dijkstra's algorithm, which was originated by computer scientist Edsger W. Dijkstra in 1956. This algorithm has many variations, however, the main function of it is to find the shortest path between any two vertices in a graph. The algorithm takes following steps in order to achieve this:

1. Mark all vertices unvisited. Create a set of all the unvisited vertices called the unvisited set.
2. Assign to every vertex a tentative distance value (called such as it is not the final distance value): set it to zero for the initial vertex and infinity for all other vertices. Set the initial vertex as current.
3. For the current vertex, consider all of its unvisited neighbours (adjacent vertices) and calculate their tentative distances through the current vertex. Compare the newly calculated tentative distance to the current assigned value (if any) and assign the smaller one to the vertex.
4. When all of the unvisited neighbours to the current vertex have been considered, mark the current vertex as visited and remove it from the unvisited set. (A visited vertex will never be checked again)
5. If the destination vertex has been marked visited, then stop. (for the shortest path between two vertices)
6. Otherwise, select the unvisited vertex that is marked with the smallest tentative distance, set it as the new "current vertex", and repeat step three onwards.

These steps, which are often shown in a flow chart, find the shortest path (the path with the lowest cost) from a source vertex to a destination vertex. In the instance of Google Maps, the graphs are huge and so looking at the whole graph would be very time consuming, so the algorithm must prioritise the closest vertices and edges (roads and junctions in this case) and consider them first. It does this by using a min-priority queue, which orders data in such a way that reduces computing time. It gives each element an associated priority, resulting in the highest priority elements being served first. In the case of Google Maps, the highest priority elements would be the fastest roads when looking for the fastest path from one place to another.

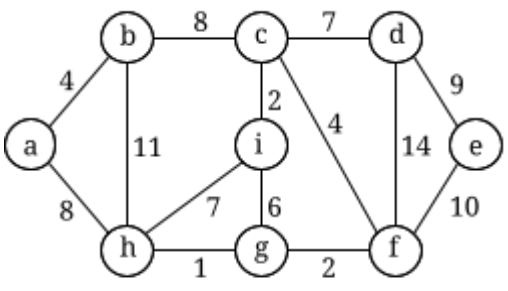

Figure 6: Example Weighted Graph

Vertex	Order of labelling	Final label
Working values		

Figure 7: Vertex Box

Using the undirected graph (Figure 6) as an example of a small road network, with the roads modelled as weighted edges and and the junctions as vertices, let us find the shortest path from vertex a to vertex e. The weight associated with each individual edge in this case is the time it takes to traverse the road. To use Dijkstra's algorithm by hand, it is useful to use a table (Figure 7) to note certain information for each vertex. Although, when used in computer applications, this process would be completed in a different format, the principles are the same. Initially, the set of unvisited vertices for this graph is $\{a, b, c, d, e, f, g, h, i\}$. The source or start vertex, labelled a, is considered first and its distance value is set to zero, whilst that of all other vertices is set to infinite. The adjacent vertices to the current vertex, a, are then considered. These vertices are b and h. For b, the tentative distance through the current vertex, a, is $0 + 4 = 4$. As this is the first value calculated for b, it is therefore also the smallest and so is assigned to the working values section of the table. For the other adjacent vertex, h, the tentative distance is $0 + 8 = 8$. All the unvisited neighbours to the

current vertex have now been considered, so a is marked as visited and removed from the unvisited set. The destination vertex has not been reached, and so the next vertex with the smallest working value, b, is selected and set as the new current vertex. The process is then repeated from step three onwards, starting with the adjacent vectors to b, until a shortest path is found from a to e, and this essentially forms the basis of these navigation applications.

To think that all these advancements, that form the foundations of systems and applications that are now so integrated into our everyday lives, all stemmed from great mathematicians such as Leonard Euler, and his solution to one seemingly small problem, is a tiny glimpse into the wonders of what mathematics is capable of achieving.

References:

Marcus du Sautoy, "A Brief History of Mathematics". BBC 4, 25 June 2010, https://www.bbc.co.uk/programmes/b00srz5b/episodes/downloads

"The Seven Bridges of Königsberg ". https://www.youtube.com/watch?v=W18FDEA1jRQ

Carlson, Stephan C.. "Graph theory". Encyclopedia Britannica, 24 Nov. 2020, https://www.britannica.com/topic/graph-theory.

Barnett, Janet Heine, "Early Writings on Graph Theory: Euler Circuits and The Konigsberg Bridge Problem". Colorado State University, 8 December 2005, http://www-users.math.umn.edu/~reiner/Classes/Königsberg.pdf.

Deo, Narsingh, "Graph Theory with Applications to Engineering and Computer Science ". Computer Science Department, Washington State University, 1974, https://www.edutechlearners.com/download/Graphtheory.pdf

Dooren, Paul Van, "Graph Theory Applications". Université catholique de Louvain, August 2009, http://www.hamilton.ie/ollie/Downloads/Graph.pdf

"Topology Underground". Irish Times, 17 January 2013, https://thatsmaths.com/2013/01/17/topology-underground/

"Topics in Classical Graph Theory". https://nptel.ac.in/content/storage2/courses/111104026/lecture39.pdf

"Planar Graphs". https://web.iitd.ac.in/~bspanda/planar.pdf

"Glossary of Graph Theory". Wikipedia, 2 March 2021, https://en.wikipedia.org/wiki/Glossary_of_graph_theory

"Dijkstra's Algorithm". https://www.youtube.com/watch?v=GazC3A4OQTE

"Dijkstra's Algorithm ". Wikipedia, 26 March 2021, https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

Figure 1: 'Solutio problematis ad geometriam situs pertinentis', Eneström 53, MAA Euler Archive, accessed 26 March 2021, <http://eulerarchive.maa.org/>

Figure 3: Tube Map 2021, Transport for London, accessed 26 March 2021, <https://tfl.gov.uk/maps/track/tube>