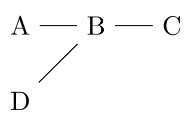
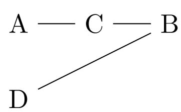
Graph Theory Report

Isak Oswald

March 24, 2025

1 Graph Examples

1. 

2. 

2 Introduction

Graphs are a collection of pretty simple concepts, however, this content covers a-lot of content.

3 Basic Graphs.

3.1 What is a graph?

A graph is a pretty simple concept as it is tightly related to sets. As a set is just a collection of elements, as a graph is made up of a collection of objects which are called nodes or vertices. Similarly, the relationships connecting these objects are called edges. This can be seen in the "example" section above. There are many different types of graphs which we will cover in more depth later, however, some common ones to look out for are labeled and unlabelled graphs (where the vertices are labeled or unlabelled), weighted graphs (the edges have a weight), and directional graphs (the edges have a specified direction). Graphs can be expressed in two main ways, through drawing (as shown above), and in a more mathematical, rigorous fashion called set-builder notation. The same graph above would look like this in set-builder notation: $V = \{A, B, C, D\}$, $E = \{\{a, b\}, \{b, c\}, \{b, d\}\}$ Where V is a set of vertices and E is a group of sets which are the edges. The formal definition of graph states that "A graph is an ordered pair $G = (V, E)$ where V is a non-empty set and E, and is a two-element subset of B.

3.2 Graph equality and Isomorphism

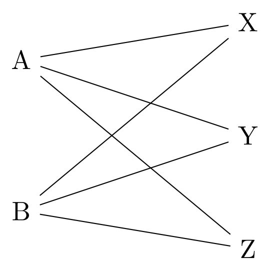
Two graphs are equal if $G_1 = (V_1, E_1)$ is the SAME as $G_2 = (V_2, E_2)$ Let's take a look at two graphs (graph 1 and 2 in the "example" section). We can see that $G_1 \neq G_2$. A similar concept relating to equality is Isomorphism, a graph that is Isomorphic means that it can have its vertices re-mapped with a bijective function. As seen in graphs 1 and 2, they are indeed isomorphic as if we remap G_2 they will be the same, for example, $f(X), f(C) = B, F(B) = C$ which are the same graph.

3.3 Foundational properties of graphs

Graphs have many different features to look out for, one important one is being connected. A graph is said to be connected if you can travel from any one vertices to any other vertices on the graph. If we look back to "example" graph 1, we can see that if you start at vertices D, you can walk to any other vertices by following a set of edges. Secondly, a graph can contain a cycle which means that a graph starts and ends at a vertex, if we look back at example graph one, we can see it DOES NOT contain a cycle, however if vertices D were connected to vertices A with an edge, it would then contain a cycle. Imagine you were a hiker walking along these vertices and edges, in "example" graph 1, you would not be able to walk in a continuous loop, you would have to retrace your steps. Every vertices on a graph contains something called its degree, its degree refers to how many edges exit the specific vertices, for example, in "example" graph 1, we see that A has a degree of one, and B however has a degree of 3 as it has 3 edges "going out" from the node. Graphs can also contain sub-graphs which means V' is a subset of V and E' is a subset of E, this is just like a subset of sets, however, the set objects are then connected by edges. As we have now explored graphs and subgraphs, we can then look at something called an induced subgraph which is a similar but slightly different concept. A sub-graph H is an induced subgraph of G if it contains all the edges between its selected vertices exactly as they appear in G which can be easily visualised if drawn out.

3.4 Named graphs

Some graphs are so common that they have their names to be easily referred to. Firstly, we have the complete graph which is denoted as K_n which is the complete graph of n vertices. A complete graph refers to every vertices being connected. We have something called bipartite graphs which are denoted as K_{mn} which m and n are the sets of vertices. A graph is bipartite if it can be "split into two" where each side doesn't have any connections between the distinct sets (example below). We then have a graph containing a cycle which is denoted as C_n with n being the number of vertices. We explained cyclic graphs (meaning containing a cycle) before so no need to do it again. Lastly, we have a path graph which is denoted as P_{n+1} where $n + 1$ represents vertices (which n just represent edges) of a graph. An example of a bipartite graph can be seen below (as this is the most confusing to explain in words)



As we can see, this graph is split into two distinct halves (so it can be split up into two sets) without any vertices being connected on each side.

3.5 Handshake lemma

Put simply (as this is a personal reflection to come back to), the handshake lemma states that the sum of all vertices degrees is ALWAYS double the number of edges. For example, if someone gave us the set of degrees $(4, 4, 2, 4, 2)$ we know we would have 8 vertices (because $16/2$ is 8). Look up example graphs to see why this works (each edge represents a one degree for each node)

4 More advanced topics

4.1 Trees

A tree means that the graph is connected, however, has no cycle. A perfect example of a tree would be "example" graph 1. We can see that it has no cycle, however is still connected (you can travel from any one node to any other node). However, it may be hard to distinguish if something is a tree which is why some properties have been created to highlight what you can and cannot do regarding trees. One property states that "a graph is a tree if and only if between every pair of distinct vertices D, there is a unique path." If we look back to the example graph one, we see that this holds. A second important property is that any tree with AT LEAST two vertices has AT LEAST two vertices with a degree of one (the starting and ending vertices). This can also be expressed as $e = v - 1$ where e represents edges and v represents vertices. We can also see that this holds for our graph one as A and D have a degree of one (also C too in this case). Here we have just covered a basic tree, however, there are also other types of trees including a rooted tree and a spanning tree. A rooted tree is when a vertex is specified as the root which creates a hierarchal structure. One example of a rooted tree we are almost all familiar with is a family tree, similar to a family tree, we also refer to vertices as siblings, parents, grandparents, etc depending on their relationships with other nodes. On the other hand, a spanning tree is just a subgraph of a graph that includes all vertices of G, G is connected (there is a path between every pair of vertices), and contains no cycles (it is a tree). Here is an easy algorithm to find a spanning tree (note, every connected graph has a spanning tree, even if it is ALL READY a spanning tree). First, remove the edge that completes the cycle and if there are multiple cycles, repeat the process until all cycles are removed while keeping the graph connected.

4.2 Forest

Trees and forests are closely related, similar to how we think about them in real life. If we ask ourselves "What is a forest" you would likely say "Just a lot of trees" which is exactly right, a forest is just a group of trees. There is one main property to help us distinguish between what is a forest and what is not, the property states that "a graph is a forest if and only if between any part of vertices, there is a maximum of one edge". If we were to group "example" graph 1 and example graph two in one big circle it would be a forest (in this case there are no edges connecting them).

4.3 Walks, trails, and paths

A walk is the simplest of the three concepts as it is just a sequence of vertices that you follow. Lets look at "example" 1 again and imagine you are a hiker walking along the edges which represent the walking path. A valid walk would be A, B, A, B, A, B, C, this process is a walk from $A- > C$, similar to walking on a hiking track, you can walk back and forth as much as you want. Secondly, a trail is just a walk with no repeated edges, therefor A, B, A, B, A, B, C would not be a valid trail as it has repeated edges, a valid trail from $A- > C$ would look like A, B, C (note no repeated vertices.). Thirdly, a path is a trail where no vertices or edges are repeated. A trail from $A- > C$ denoted as A, B, C would also be a valid trail.

5 More advanced types of graphs + Euler's formula

5.1 Planar graphs

A planar graph is a graph that can be drawn without ANY edge crossing or overlapping, one example of a NON planar graph (based on how its currently drawn) would be the bipartite graph shown in a previous example. We can see that this is a nonplanar graph because it had edges crossing. You might ask, "Well how do we tell if a graph could be represented in a planar way without actually doing it", this is where Euler's formula and an important inequality come in, but first, let talk talk more about planar graphs in general. As a planar graph has no edges overlapping, it can be represented as a 3D object on the plane, this means that instead of ONLY having vertices and edges, it now has vertices, edges, and faces. A face at minimum has to be enclosed by

3 edges for it to be a valid face, however, these edges that make up a face also are a part of the second face (because if you made one face which is at minimum a triangle with 3 edges, you would only need an additional two edges to create a second face, not three) based on our understanding here, we can use the equality $2e \geq 3f$ where e is edges and f is faces. This represents a relationship between the amount of edges and amount of possible faces. Euler's formula also portrays a key relationship between edges, vertices and faces, this formula is: $v - e + f = 2$. I'm not going to get into the proofs or why they work here in a summary, however, just know they do (refer to the textbook if needed). Now, to answer our old question "well how do we tell if a graph could be represented in a planar way without actually doing it", we can use our new fancy inequalities and formula to solve this without actually having to draw it. Just say we are giving a graph with 4 vertices, 6 edges, and 4 faces, this would translate to $4 - 6 + 4 = 2$ which is indeed true, therefor we are mostly sure it can be represented as a planar graph (except for some advanced graphs which are out of the scope of this content, however, I did do some research myself). We can also use our inequality to check if the graph *is* planar, this would look like, $2(6) \geq 3(4)$ which still holds, so looks good! Sometimes, there are too many edges and too few vertices which means that there will be some overlap of edges. When looking back at our named graphs, the first complete graph where a planar representation is not possible is K_5 which can be tested using our formula and inequality.

5.2 Polyhedra

A polyhedra simply refers to a three-dimensional object made up of flat polygonal faces, straight edges, and vertices (such as a cube or a pyramid). A convex polyhedra contains its edges inside the object, which again could be a cube. Due to still practising LaTex, I am unable to find out a way how to draw these, therefore refer to my notes attached if needed.

6 Euler path and Euler circuits.

6.1 Euler path

Put simply, an Euler path is just a walk through a graph that uses EVERY edge exactly one time. To determine whether a graph has an Euler path, we can check the degree of each node within the graph. If more than two nodes have an odd degree, it means that the graph CANNOT contain an Euler path. For a graph to have an Euler path it must also satisfy these conditions: The graph must be connected (there should be a path between any two vertices) and at most two vertices can have an odd degree.

6.2 Euler Circuit

Similarly to a Euler path (considering the same conditions) a graph can contain a Euler circuit if the degree of EVERY vertices has an even degree. Fleury's algorithm is a way to find the Euler circuit explained almost perfectly on Deakin's Cloud Deakin website. The algorithm is as follows:

"Input: An Eulerian graph G.
Choose a starting vertex of odd degree if one exists. Go to any available edge, choosing a bridge only if there is no alternative. Then record that edge, erase the edge and any isolated vertex. Repeat step 2 until there are no more edges.
" (Deakin University, n.d.)

Similar to an Euler path, a Hamiltonian path is almost identical. As discussed previously, an Euler path is concerned with visiting all edges of a graph without repetition, the Hamiltonian path's only difference is that it is concerned with visiting all of the vertices of a graph. Similar to the Euler path conditions of being a complete graph, the Hamiltonian path is only valid if we are conducting it on a complete graph (for example it cant have isolated vertices).

7 References

Deakin University. (n.d.). Deakin. https://www.deakin.edu.au/