Victoria University of Wellington
School of Mathematics and Statistics

Graph Theory MATH 361, T1 2024

1. The Basics

A *graph* consists of a set V of *vertices*, a set E of *edges*, and an *incidence function* that maps each edge in E to either one or two vertices in V.

We say "one vertex" but "two vertices". To say that $G = (V, E)$ is a graph means that G is a graph with vertex set V and edge set E. Note that the incidence function is implicit: we say that an edge $e \in E$ is *incident* with a vertex $v \in V$ when v is one of the vertices that the incidence function maps e to. Given a graph G where the vertex and edge set have not been specified, we use the notation $V(G)$ to refer to the vertex set, and $E(G)$ to refer to the edge set.

It is possible to have graphs with an infinite number of vertices or edges, but, in MATH361, all graphs will have a *finite* number of vertices and edges.

Intuitively we think of the vertices as points and the edges as curves joining the points that they are incident to. (What about when an edge is only incident to one vertex? We'll come back to that in a moment.)

Graphs provide mathematical models of many, many real world situations. For example, electricity grids, road maps, wiring diagrams, our brain, directory structures in a computer, a computer network, flight routes, the internet, an exam timetable, a map of the world, can all be thought of as graphs — and these are just a few examples; there are heaps more.

- A nice thing about graphs is that we can draw pictures of them. This will be vital to helping us think about graphs, but there is a danger. Be aware that the same graph can have many different drawings which look quite different. A graph only tells us which edges are incident with which vertices. It tells us *nothing* about distance, angle, shape etc.
- It's nice if you can draw a graph so that edges only cross at vertices, but sometimes you cannot. Such is life. When you draw a graph, take care to make your vertices clear, so that crossings are not confused with vertices.

A *loop* is an edge that is incident with only one vertex. When drawing a graph with a loop, we draw the loop as a closed curve that starts and ends at the one vertex it is incident to. Two distinct non-loop edges are *parallel* if they are incident with the same pair of vertices. A *simple graph* is one that has no loops or parallel edges. Simple graphs can be complicated! In a simple graph, we can uniquely describe an edge by the pair of vertices it is incident to, so in this case it is often to convenient to think of an edge as a pair of vertices.

Recall that a graph is equipped with an incidence function that tells us what vertices are *incident* to each edge. When e is an edge of a graph that is incident to the vertices u and v, we say that u and v are the *ends* of e; note that $u = v$ when e is a loop, otherwise $u \neq v$. When e is a non-loop edge, we often use the shorthand $e = uv$ to mean that u and v are the ends of e. For a set of edges F in a graph, the vertices *incident with* F is the union of the vertices incident with each edge in F. We say two vertices u and v are *adjacent* if there exists an edge e such that e is incident to u and v, otherwise they are *non-adjacent*. We also say two edges e and f are *adjacent* if there exists a vertex v such that e and f are both incident to v.

Warning: Graph terminology is *not* standard. Graph theory is a young subject and has only been pursued seriously since the 20th century. This means that the dust has yet to settle on terminology. If you read texts on graphs, you need to make sure that you understand their terminology first. For example, what we call a graph, many authors call a multigraph.

Walks, paths, and cycles. A *walk* in a graph is an alternating sequence of vertices and edges

$$v_1, e_1, v_2, e_2, \ldots, e_{n-1}, v_n$$

such that each e_i is incident with both v_i and v_{i+1}.

A *path* is a walk in which no vertex appears more than once. The *length* of a walk (or path) is the number of edges in the walk (or path).

If the first and last vertex of a walk are equal, then it is a *closed walk*.

Take a closed walk with length at least one, in which only the first and last vertex are equal. The edges and vertices of such a closed walk form a *cycle*. Note that many different walks can describe the same cycle. (Why?)

Nag: The above structures are fundamental in graphs and if you don't take the trouble to learn exactly which is which, then understanding the lectures will become very difficult. This is essentially true of all the definitions we will encounter in graph theory.

The following lemma is straightforward to prove; so much so that its proof is left as an exercise. Nonetheless, it is saying something, and it's worth internalising what that is. (One way to do this is by having a go at proving it yourself!)

Lemma 1.1. *Let G be a graph with a walk W from a vertex u to a vertex v. Then there is a path from u to v that uses a subset of the edges of W.*

Exercise 1.2. *Prove Lemma 1.1.*

Matrices associated with graphs. Let $G = (V, E)$ be a graph, where $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{e_1, e_2, \dots, e_m\}$. Then the *incidence matrix* of G is an $n \times m$ matrix whose rows are labelled by V; columns are labelled by E; and the entry in the row labelled v_i and column labelled e_j is 1 if v_i is incident with e_j, otherwise it is 0.

On the other hand, the *adjacency matrix* of G is the matrix whose rows and columns are labelled by V where the entry in the row labelled v_i and column labelled v_j is 1 if v_i and v_j are adjacent (i.e. there is an edge joining v_i and v_j), otherwise it is 0.

→ For *human* understanding, the best way to describe a graph is by a picture; for *computers*, the best way is by a matrix. This contrast occurs throughout mathematics. For our own understanding, we need a way of seeing structures intuitively, but for computation and precision we also need a formal way of describing structures.

Vertex degree. The *degree* of a vertex v is the number of edges incident with it. The degree of v is denoted $d(v)$. Note that, by convention, a loop contributes 2 to the degree of a vertex.

Recall that if S is a set, then $|S|$ denotes the number of elements in S, that is, the *size* or *cardinality* of S.

Theorem 1.3 (The Handshaking Lemma)**.** *In a graph $G = (V, E)$, the sum of the degrees of the vertices is twice the number of edges. More precisely,*

$$\sum_{v \in V} d(v) = 2|E|.$$

Proof. To prove the Handshaking Lemma we use the *modified* incidence matrix where the entry in the v_i row and e_j column is a 2 if e_j is incident with v_i and v_i is a loop, otherwise it is the same as in the standard incidence matrix.

Consider the graph $G = (V, E)$ and let M be the modified incidence matrix of G. Let S be the sum of all the entries in M. We will find S in two different ways.

The sum of the entries in each *column* of M is 2. There is one column for each edge. Thus we get

$$S = 2|E|.$$

Consider the rows. The sum of the entries in a row labelled by a vertex v is $d(v)$ as the non-zero entries correspond to edges incident with v. Thus we get

$$S = \sum_{v \in V} d(v).$$

This shows that $\sum_{v \in V} d(v) = S = 2|E|$, as required. $\square$

Don't think that just because the proof is short that it is uninteresting. It used the technique of showing that two things were equal because they *counted the same thing in different ways.*

Corollary 1.4. *Every graph has an even number of vertices of odd degree.*

Proof. Towards a contradiction, suppose that there exists a graph G that has an odd number of vertices of odd degree. Then the sum of the degrees, $\sum_{v \in V(G)} d(v)$, is an odd number. But, by the Handshaking Lemma, this number is twice the number of edges in G, which is an even number. This contradiction shows that our initial assumption is false; that is, every graph has an even number of vertices of odd degree. $\square$

Subgraphs. Let $G = (V, E)$ be a graph, and let $V' \subseteq V$ and $E' \subseteq E$ such that the vertices incident with E' are contained in V'. Then $G' = (V', E')$ is a *subgraph* of G.

We can also consider subgraphs that are determined ("induced") by just the vertices. Let $U \subseteq V$. The subgraph of G *induced* by U, denoted $G[U]$, has vertex set U and edge set consisting of the edges of G whose ends are contained in U. (Note that, for a non-loop edge $e = vw$, *both* v and w must be in U for the edge e to appear in $G[U]$.) We say that $G[U]$ is an *induced subgraph* of G.

We could also consider subgraphs induced by a set of edges. Let $F \subseteq E$. The subgraph of G *induced* by F, denoted $G[F]$, has edge set F and vertex set consisting of all vertices of G incident with at least one edge in F. We say that $G[F]$ is an *edge-induced subgraph* of G.

For clarity, we could call an “induced subgraph” a “vertex-induced subgraph”, but it is standard practice to assume an “induced subgraph” is vertex induced unless specified otherwise.

Remember how we said a cycle was a closed path where only the first and last vertex are equal? From this perspective, a cycle of length t is an sequence $v_1, e_1, v_2, e_2, \ldots, v_t, e_t, v_1$ that alternates between vertices and edges. Often, it is instead more useful to think of such a cycle as a subgraph on vertex set $\{v_1, \ldots, v_t\}$ and edge set $\{e_1, \ldots, e_t\}$. Since a cycle is uniquely determined by its edge set, other times it is more convenient to think of a cycle as a set of edges. Although context will usually help, try to make clear if you are using this viewpoint. Most commonly we will view a cycle as a subgraph.

Similarly, sometimes it is convenient to view a path $v_1, e_1, v_2, e_2, \ldots, v_t, e_t, v_{t+1}$ as a subgraph with vertex set $\{v_1, \ldots, v_t\}$ and edge set $\{e_1, \ldots, e_t\}$, or simply as a set of edges $\{e_1, \ldots, e_t\}$.

Connectivity. For any graph G with vertex set V, we now define a relation $\sim$ on V as follows: for u and v in V, we have $u \sim v$ if and only if there is a walk from u to v.

Theorem 1.5. *For any graph G with vertex set V, the relation $\sim$ is an equivalence relation on V.*

Proof. [redacted] $\square$

Recall that associated with *any* equivalence relation $\sim$ on a set S, there is a partition of S into subsets called *equivalence classes*. Members of the same equivalence class are all related to each other.

Thus, for our equivalence relation $\sim$, there is a partition of V into equivalence classes. Moreover, if V_i is an equivalence class of $\sim$, then there is a walk (and hence, by Lemma 1.1, a path) between any pair of vertices in V_i, while there is no walk (or path) between a vertex in V_i and a vertex outside V_i.

Suppose that for a graph G, there are t equivalence classes of $\sim$, and these are $V_1, V_2, \ldots, V_t$. Then $G[V_1]$, $G[V_2]$, …, $G[V_t]$ are called the *components* of G. We have defined a component C to be an (induced) subgraph, so it is a graph in its own right, with vertex set $V(C)$ and edge set $E(C)$.

→ Every edge of G belongs to precisely one of the components. So not only can we refer to the component that a vertex of G belongs to, but we can unambiguously refer to the component that an edge of G belongs to.

A graph is *connected* if it has exactly one component. Equivalently, a graph G is connected if it has at least one vertex and there is a walk between every pair of vertices in G. We say that a graph is *disconnected* if it has at least two components.

A component C of a graph G is a maximal connected subgraph: there is no connected subgraph of G whose vertex set contains $V(C)$ and edge set contains $E(C)$ having more vertices or edges than C.

Often in mathematics, there is a trade-off between precision and intuition. When first learning concepts, intuition is potentially more important. But in order to really know something is true, we require precision in our proofs. In these notes, we'll use a box, like below, when we are wearing a "pedantic" hat, rather than a "intuitive" hat. Perhaps on first read of these notes, these might not seem so important, whereas on a later read, you might appreciate the attention to detail.

We say that $G = (\emptyset, \emptyset)$ is the *empty graph*, and a graph is *non-empty* if it has at least one vertex. Since the empty graph has zero components, under our definitions it is neither connected nor disconnected! On the other hand, any non-empty graph is either connected or disconnected (and not both).

Complete graphs. The simple graph with n vertices and all possible edges is called the *complete graph on n vertices* and is denoted by K_n.

Theorem 1.6. *For each positive integer n, the complete graph K_n has $n(n-1)/2$ edges.*

Proof. [redacted] □

Can you draw K_4 without edges crossing? What about K_5?

Bipartite graphs. A graph $G = (V, E)$ is *bipartite* if its vertex set V can be partitioned into two sets V_1 and V_2 such that each edge of G joins a vertex in V_1 to a vertex in V_2.

→ There is a potential trap in the above definition, due to the phrase "*can be* partitioned". Think about it!

The *length* of a cycle is the number of edges in the cycle. Since the number of vertices in a cycle is the same as the number of edges (why?), we could equivalently have said that the length is the number of vertices in the cycle. (Note the difference with paths: there we must count the edges.)

Theorem 1.7. *A graph G is bipartite if and only if G has no cycles of odd length.*

This theorem describes a fundamental property of bipartite graphs, but at the moment we lack tools to help us prove it. We will build up some more theory regarding trees, before proving this result in Section 2.

The *complete bipartite graph* $K_{s,t}$ has vertex set $V_1 \cup V_2$, where $|V_1| = s, |V_2| = t$, the sets V_1 and V_2 are disjoint, and there is a single edge between each vertex in V_1 and each vertex in V_2.

Can you draw $K_{2,5}$ without edges crossing? What about $K_{3,3}$?

Theorem 1.8. *For any positive integers $s, t \geq 1$, the complete bipartite graph $K_{s,t}$ has st edges.*

Proof. The graph $K_{s,t}$ has s vertices of degree t and it has t vertices of degree s. Therefore the sum of the degrees is $st + st = 2st$. It now follows from the Handshaking Lemma that $K_{s,t}$ has st edges. □

Isomorphism. Previously, we said K_n was *the* simple graph with n vertices and all possible edges. But consider K_3. In fact there are an infinite number of simple graphs with 3 vertices and all possible edges, having different vertex sets.

Do we care? The point is that all of the different graphs with 3 vertices and all possible edges, are "structurally the same", or "the same apart from labelling", and so it is likely that you naturally, and correctly, focussed on the structural property, rather than the labelling. We need to make this precise.

Throughout mathematics, the word we use to say "these two objects are structurally the same, but possibly labelled in different ways" is *isomorphic.* What this means depends on the type of structure we are talking about.

Recall that a function $f : S \to T$ is a *bijection* if it is both injective (one-to-one) and surjective (onto); that is, for each member t of T, there is *exactly one* member s of S such that $f(s) = t$.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs. Then an *isomorphism* between G_1 and G_2 is a pair of bijections $f : V_1 \to V_2$ and $g : E_1 \to E_2$ such that e is incident with v in G_1 if and only if $f(e)$ is incident with $g(e)$ in G_2.

The graphs G_1 and G_2 are *isomorphic* if there exists an isomorphism between G_1 and G_2; we denote this as $G_1 \cong G_2$.

So what is K_n? It is the name for any simple graph isomorphic to one obtained by choosing n vertices and adding all possible edges between them.

Notions of substructure. What does it mean for one graph to "contain" another? The answer depends on the notion of substructure that we choose. We have already seen two different notions of substructure, which we recall in a moment.

We first consider what it means to delete an edge or vertex from a graph. If e is an edge of a graph G, then $G \backslash e$ ("G *delete* e") is the graph obtained by deleting the edge e and leaving all other edges and vertices intact. If v is a vertex of G, then $G - v$ is the graph obtained from G by removing v and all edges incident with v. (Note: when we delete a vertex, we must also delete all edges incident with it.)

Now let G and H be graphs. Then H is a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Equivalently, H is a subgraph of G if H can be obtained from G by deleting edges and vertices. On the other hand, H is an *induced subgraph* of G if H can be obtained from G by deleting vertices (only).

Note the difference: every induced subgraph is a subgraph, but not every subgraph is an induced subgraph. For example, let e be an edge of K_5, so $K_5 \backslash e$ is obtained from K_5 by removing a single edge. Then $K_5 \backslash e$ is a subgraph of K_5, but not an induced subgraph.

What are the induced subgraphs of K_5?

There is one more important notion of substructure that we will see: the notion of a *minor*. First, we recall what it means to contract an edge.

Contraction and minors. We already know that if e is an edge of G, then the deletion of e from G, denoted $G \backslash e$, is the graph obtained by removing the edge e (only).

There is another fundamental way to remove an edge from a graph. We first give the formal definition. Recall that for sets X and Y, we use $X \setminus Y$ to denote *set difference*, i.e. the set of elements in X that are not also in Y.

> Let $G = (V, E)$ be a graph, and let $e = uv$ be an edge of G. The graph G *contract* e, denoted G/e, is obtained as follows. The vertices of G/e are $(V \setminus \{u, v\}) \cup \{w\}$, where w is not in V. The edges of G/e are $E \setminus \{e\}$. We can describe the incidence relation as follows: for $f \in E \setminus \{e\}$,
>
> - if f is incident with either u or v in G, then f is incident with w in G/e,
> - if f is incident with $z \in V \setminus \{u, v\}$ in G, then f is incident with z in G/e.

A formal definition like this is useful in proofs because of its precision, but for human understanding we need another way. To visualise contraction imagine the edge $e = uv$ shrinking until it squeezes the two vertices u and v together into a new vertex which, in the above definition, we call w.

→ Often, when contracting an edge, we are only interested in the isomorphism class, so the name given to the vertex resulting from the contraction (i.e., w in the above definition) is not important. This is also evident in the notation G/e. When the name given to this vertex is important, be explicit!

We say H is a *minor* of a graph G if H can be obtained from G by a (possibly empty) sequence of edge deletions, edge contractions, and vertex deletions.

Minors will be important to us throughout this course. While edge deletion is straightforward, contraction is more subtle. In what follows we start to develop some basic properties of contraction.

Properties of contraction. Contracting an edge $e = uv$ can sometimes seem to significantly change the appearance of a graph, but the next lemma shows that the parts of the graph that do not contain both u and v remain essentially the same.

Lemma 1.9. *Let $G = (V, E)$ be a graph, and let $e = uv$ be a non-loop edge of G.*

(i) *Let U be a subset of V that contains at most one element of $\{u, v\}$. Then $G[U]$ is isomorphic to a subgraph of $(G/e)[U]$.*

(ii) *Let F be a subset of E that is incident with at most one element of $\{u, v\}$ in G. Then $G[F] \cong (G/e)[F]$.*

Proof. Let w be the vertex resulting from the contraction of e.

Consider (i). The induced subgraph $G[U]$ contains at most one of u and v. Let H be the graph obtained from $G[U]$ by relabelling either u or v to w, if such a vertex appears in $G[U]$; otherwise let $H = G[U]$. By the definition of contraction, if e is an edge of H, then e is also an edge of $(G/e)[U]$, which is incident to the same pair of vertices, except that u or v is replaced with w. Now (i) follows.

Similarly, for (ii), the edge-induced subgraph $G[F]$ contains at most one of u and v. By the definition of contraction, apart from relabelling either u or v to w, the edge-vertex incidences do not change, so that the structure does not change. □

→ The upshot of the last lemma is: the structures in a graph that change in the contraction are those that contain both u and v.

Exercise 1.10. *Show that "is isomorphic to a subgraph of" in Lemma 1.9(i) cannot be replaced with "is isomorphic to". In other words, give a counterexample to the following statement: if U is a subset of V that contains at most one element of $\{u, v\}$, then $G[U] \cong (G/e)[U]$.*

Why did we need to use isomorphism in Lemma 1.9, e.g. can we not replace '$\cong$' with '$=$' in (ii)? The reason we might not have equality is that we give the new vertex, resulting from the contraction, a label different from $\{u, v\}$. In the case that U (or F) contains (or is incident with) u, then if we relabelled the new vertex resulting from the contraction as u, the graphs would indeed be equal.

What if we try to contract a loop?

Observation 1.11. *Let G be a graph with a loop e. Then $G/e \cong G\backslash e$.*

We leave this as an observation (with no proof), as it is a direct consequence of the definition of contraction.

Lemma 1.12. *Let G be a graph with a non-loop edge $e = uv$, and let C be a subset of $E(G) \setminus \{e\}$. Then C is the set of edges of a cycle in G/e if and only if either*

(i) $C \cup \{e\}$ *is the set of edges of a cycle in G, or*
(ii) C *is the set of edges of a cycle in G, and the vertices of this cycle contain at most one element of $\{u, v\}$.*

Proof. ($\Leftarrow$) Assume that (i) holds, so $C \cup \{e\}$ is the set of edges of a cycle in G. Then C is the set of edges of a path from u to v in G. Say $C = u, e_1, \ldots, e_n, v$. Let w be the vertex resulting from the contraction of e in G/e. Then $w, e_1, \ldots, e_n, w$ is a cycle in G/e, and the set of edges of this cycle is C.

Now assume that (ii) holds. By Lemma 1.9(ii), $G[C] \cong (G/e)[C]$, so C is the set of edges of a cycle in G/e. This proves one direction.

($\Rightarrow$) Conversely, assume that C is the set of edges of a cycle in G/e, and w is the vertex resulting from the contraction of e. If w is not a vertex in $(G/e)[C]$, then $G[C]$ is a cycle that does not contain u or v, so $G[C] \cong (G/e)[C]$ by Lemma 1.9(ii), and (ii) holds. If w is a vertex of $(G/e)[C]$ but $G[C]$ only contains one of u or v, then, similarly, (ii) holds. If w is a vertex of $(G/e)[C]$ and $G[C]$ contains both u and v, then C is the set of edges of a path from u to v in G. Then $C \cup \{e\}$ is the set of edges of a cycle in G, so (i) holds. □

Lemma 1.12 only concerns when e is not a loop – so what happens to cycles when contracting a loop? We noted, as Observation 1.11, that when e is a loop, G/e is just the same as $G \backslash e$ (up to the label of the vertex incident to e). So the cycles of G are just the cycles of G/e plus the extra cycle of length one on the edge e.