Lecture 4: Graph Theory: Introduction and Terminology

Eric Vigoda
Discrete Mathematics for Computer Science

January 14, 2026

We saw graphs in the first lecture. In this lecture we'll dive into graph theory.

4.1 Graphs: Definitions and Terminology

A graph consists of a set V of vertices and a collection E of edges. An edge connects a pair of vertices, thus each edge in E is a set $\{i, j\}$ where $i \in V$ and $j \in V$. A graph is represented by the pair (V, E) where V is the set of vertices and E is the collection of edges; note it is an ordered pair (V, E), it is not a set $\{V, E\}$.

We often say: "Let $G = (V, E)$ be a graph... or "For a graph $G = (V, E)$..." In this way we can use G to refer to the graph defined by the pair (V, E). Figures 1 and 2 show two example graphs where each $G = (V, E)$ has $n = |V| = 8$ vertices and $m = |E| = 9$ edges.

Throughout this course we consider simple graphs, which we simply refer to as graphs. A simple graph has no self-loops, for $v \in V$, a self-loop is an edge of the form $\{v, v\}$, and thus the edge is between a vertex v and itself. We always assume no self-loops, unless otherwise mentioned. Also, graphs have at most one edge between a pair of vertices, thus E is a set. When one wants to allow multiple edges between a pair of vertices then that is called a multigraph, and hence E is a multiset in this case; but we are not considering multigraphs in this course.

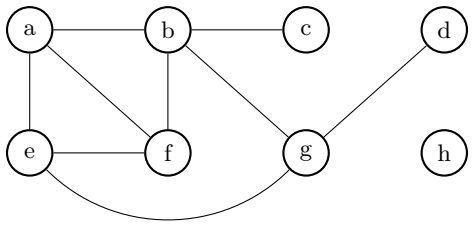

Figure 1: Example graph G_1 with 8 vertices and 9 edges.

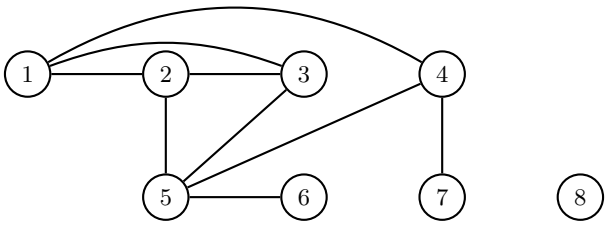

Figure 2: Another example graph G_2.

4.2 Graph Isomorphism

Consider the 2 graphs illustrated in Figures 1 and 2. For concreteness, let $G_1 = (V_1, E_1)$ denote the graph drawn in Figure 1, and let $G_2 = (V_2, E_2)$ denote the graph drawn in Figure 2.

The first graph has the following edge set:

$$E_1 = \{\{a,b\},\{a,f\},\{a,e\},\{b,c\},\{b,g\},\{g,d\},\{g,e\},\{f,b\},\{e,f\}\}.$$

And in the second graph we have the following:

$$E_2 = \{\{3,5\},\{3,2\},\{3,1\},\{5,6\},\{5,4\},\{4,7\},\{4,1\},\{2,5\},\{1,2\}\}.$$

Consider the following *bijection* $\sigma : V_1 \to V_2$ (see below for the definition of a bijection):

$$\sigma(a) = 3, \sigma(b) = 5, \sigma(c) = 6, \sigma(d) = 7, \sigma(e) = 1, \sigma(f) = 2, \sigma(g) = 4, \sigma(h) = 8.$$

Notice that the edge sets are the same under this mapping:

$$\{i,j\} \in E_1 \iff \{\sigma(i), \sigma(j)\} \in E_2.$$

Let us recall the definition of a bijection. For sets S and T, a function f with domain S and codomain T is denoted by $f : S \to T$. (Some of you may refer to the set T as the range of the function f, but technically, the term range is for the actual values obtained by the function, whereas codomain is the set of possible values.) A function $f : S \to T$ is a mapping from each $x \in S$ to an element $y \in T$. Thus, for every $x \in S$, $f(x) \in T$, see Figure 3 for several example functions. As an example, the function f_1 in Figure 3 is the following: $f_1(1) = a, f_1(2) = a, f_1(3) = c, f_1(4) = c, f_1(5) = c, f_1(6) = e, f_1(7) = f, f_1(8) = g$.

For a function $f : S \to T$, if every $y \in T$ has at least one $x \in S$ where $f(x) = y$ then we say that f is an *onto* function, it is also called a *surjective* function. If every $x \in S$ maps to a unique $y \in T$ – in other words, for all $x_1, x_2 \in S$, if $x_1 \neq x_2$ then $f(x_1) \neq f(x_2)$ – then we say f is a *one-to-one* or an *injective* function. Finally, if f is one-to-one and onto, then we say f is a **bijection**, it is also called a *one-to-one correspondence* or a *bijective* function. Note, if $|S| = |T|$ (and S and T are finite) then a one-to-one function is always a bijection.

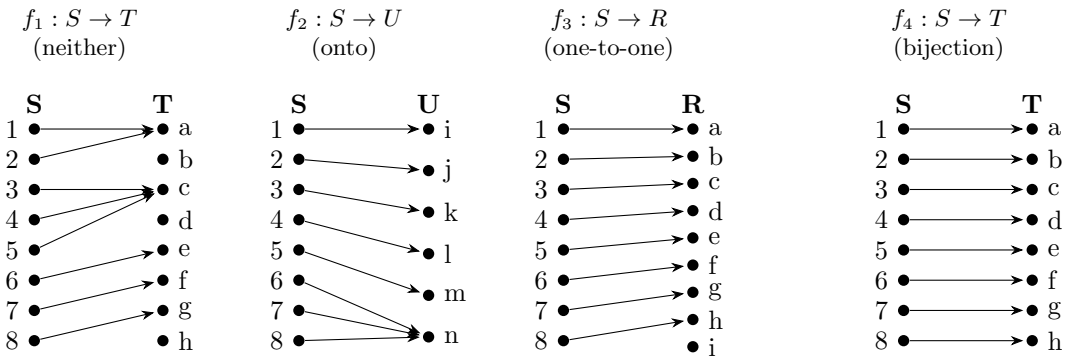

Figure 3: Examples of functions

Essentially the two graphs in Figures 1 and 2 are the same, the only difference is in the labelling of the vertices. Of course, the geometric placement of the vertices on the page is different but that is just cosmetic, a graph is defined by its edges (the pairwise interactions), and these two graphs have the same edge structure so they should be considered the same graph.

We say two graphs are **isomorphic** if there is a relabelling of the vertex sets so that the edge structures are preserved. More formally, we are looking for a bijection between the sets of vertices which preserves the edge structure.

Definition 4.1. For a pair of graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, we say the graphs G_1 and G_2 are **isomorphic** if $|V_1| = |V_2|$ and there is a bijection $f : V_1 \to V_2$ such that for all $v, w \in V_1$:

$$\{v,w\} \in E_1 \iff \{f(v), f(w)\} \in E_2.$$

If G_1 and G_2 are isomorphic, we denote it as $G_1 \cong G_2$.

Given two graphs G_1 and G_2, determining if these two graphs are isomorphic to each other is a notoriously difficult computational problem, known as the graph isomorphism problem. We will not address the graph isomorphism problem in this class, we will use the simple observation that if we relabel the names of the vertices we obtain a graph that is more convenient to refer to in our proofs and our algorithms, and this relabeling does not change the structure of the graph.

4.3 Vertex Labeling

For a graph $G = (V, E)$, let $n = |V|$. We often label the vertices V as $\{1, \ldots, n\}$ for convenience; this corresponds to a simple relabelling of the vertices, or in other words, it is a bijection between $\{1, \ldots, n\}$ and the original vertex set. In your proofs or explanations you can say: Let $V = \{1, \ldots, n\}$ or let $V = \{v_1, \ldots, v_n\}$ to make clear the labeling that you're using.

4.4 Common Graphs

Here are some common graphs, which are depicted in Figure 4:

Path graph P_n: This is a path on n vertices.

Cycle graph C_n: This is a single cycle on n vertices.

Complete graph K_n: This graph on n vertices includes all possible edges.

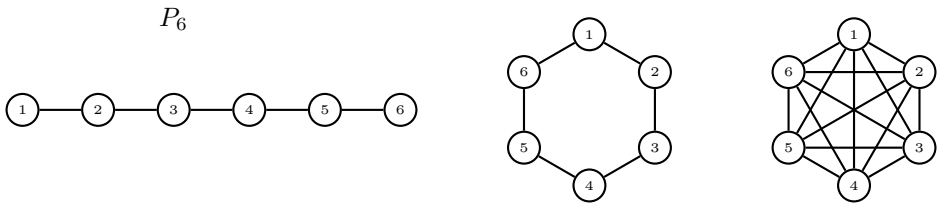

Figure 4: On the left is the path graph P_6, then the cycle graph C_6, and the complete graph K_6.

How many edges are contained in the complete graph K_n? Consider an edge $\{v, w\}$. There are n choices for vertex v. Then there are $n - 1$ choices for w, since v is not allowed. But we've double-counted as each pair is counted twice but an edge is an unordered set. Therefore, the number of edges in K_n is

$$\binom{n}{2} = \frac{n(n-1)}{2}.$$

4.5 Terminology

For an edge $e = \{i, j\} \in E$, we say i and j are the **endpoints** of edge e. And we say i and j are **adjacent** if there is an edge $\{i, j\} \in E$. Similarly, we can say a pair of edges $e, f \in E$ are adjacent if they share a common endpoint. Thus, adjacency is used for a pair of vertices which are connected by an edge, or for a pair of edges which share a common endpoint. What about between a vertex and an edge?

For an edge $e = \{i, j\} \in E$, we say e is **incident** to vertex i (and of course to vertex j as well). Then, we say the **degree** of a vertex $i \in V$ is the number of edges which are incident to it; here is a more formal definition. (I'm going to switch to referring to vertices as v and w instead of i and j so you getting used to multiple styles.)

For a vertex $v \in V$, denote the set of neighbors of v as $N(v)$ which is defined as:

$$N(v) = \{w \in V : \{v, w\} \in E\}.$$

Then, the degree of vertex v is denoted as $d(v)$ and is defined as:

$$d(v) = |N(v)|,$$

which is the number of neighbors of v or equivalently the number of edges incident to v.

As an illustration, in G_1 from Figure 1, the neighbors of vertex a are $N(a) = \{b, f, e\}$ and it has degree $d(a) = 3$. In contrast, vertex h has $N(h) = \emptyset$ and $d(h) = 0$, hence h is called an **isolated vertex**.

A **walk** is a sequence of adjacent vertices. Thus, a sequence $v_0, v_1, \ldots, v_\ell$ is a walk in $G = (V, E)$ if:

- For all $0 \leq i \leq \ell$, $v_i \in V$; and
- For all $0 \leq i < \ell$, $\{v_i, v_{i+1}\} \in E$.

If all of the vertices $v_0, \ldots, v_\ell$ are distinct (so we visit each vertex at most once) then the walk is called a **path**.

If a path starts and ends at the same vertex (i.e., $v_0 = v_\ell$) then this is a **cycle**. As an example, in the graph from Figure 1, there is a cycle a, b, g, e, a and another cycle is a, b, f. We do not consider cycles of length 2 as every edge would be a cycle in that case.

Hopefully, all of the terminology introduced above is very natural. The only less familiar terminology is that adjacency is between pairs of vertices or between pairs of edges, and thus between two objects of the same type. Whereas incidence is between a vertex and an edge, which are objects of different type.

4.6 Handshaking Lemma

The handshaking lemma relates the number of edges in a graph to the sum of the degrees. The name – handshaking lemma – comes from the example of vertices corresponding to people, and edges corresponding to pairs of people that shook hands with each other.

Lemma 4.2 (Handshaking Lemma)**.** *For any graph $G = (V, E)$,*

$$|E| = \frac{1}{2} \sum_{v \in V} d(v).$$

Proof. Fix $n = |V|$, the number of vertices, and let $V = \{1, \ldots, n\}$. We will prove the lemma by induction on $m = |E|$.

Base case: The base case is when $m = 0$ in which case $E = \emptyset$ and hence $|E| = 0$. In this case, every vertex v has degree $d(v) = 0$ and hence we also have $\sum_{v \in V} d(v) = 0$, which establishes the base case.

Inductive hypothesis: For integer $k \geq 0$, for any graph $G = (V, E)$ with $n = |V|$ vertices and $|E| = k$ edges, assume that $|E| = \frac{1}{2} \sum_{v \in V} d(v)$.

Inductive step: Now consider a graph $G = (V, E)$ with $n = |V|$ vertices and $|E| = k + 1$ edges. Consider any edge $e \in E$, and let $G' = (V, E \setminus e)$ be the graph obtained by removing edge e. Since G' has k edges, then we know by the inductive hypothesis that: $|E'| = \frac{1}{2} \sum_{v \in V} d'(v)$ where $d'(v)$ is the degree of vertex v in graph G'. Let $e = \{v_i, v_j\}$ denote the endpoints of e. Note, $d(v_i) = 1 + d'(v_i)$ and $d(v_j) = 1 + d'(v_j)$, and all other vertices have the same degree, and hence:

$$\sum_{v \in V} d(v) = 2 + \sum_{v \in V} d'(v). \tag{1}$$

Putting it all together we have the following:

$$\begin{aligned} |E| &= 1 + |E'| \\ &= 1 + \frac{1}{2} \sum_{v \in V} d'(v) && \text{by the inductive hypothesis} \\ &= \frac{1}{2} \sum_{v \in V} d(v) && \text{by Equation (1),} \end{aligned}$$

which completes the proof of the lemma. □

Remark 4.3. The above proof was meant to illustrate a proof by induction but Lemma 4.2 is easier to prove by a direct proof. Simply observe that an edge $\{v_i, v_j\} \in E$ is counted once in the degree of v_i and once in the degree of v_j. Hence, each edge is counted twice in the sum over all degrees (or each edge contributes 2 to the total sum of degrees).

As a consequence, the sum of the degrees is always even, since it's twice the number of edges. Consider the degree sequence $(1, 2, 3, 3)$; that means the graph has 4 vertices where one vertex has degree 1, one vertex has degree 2, and two vertices each have degree 3. Does there exist a graph with this degree sequence? By Lemma 4.2, the number of edges in this graph would be $|E| = \frac{1}{2}(1+2+3+3) = 4.5$, but there cannot be a fractional number of edges so there is no graph with this degree sequence.

4.7 Connected Graphs and Trees

A pair of vertices $v, w \in V$ are **connected** if there is at least one path between v and w. Note, the property of being connected is symmetric: since the graph is undirected, if there is a path from v to w then this same path is also from w to v. Connectedness is also transitive: if vertices v_i and v_j are connected, and vertices v_j and v_k are connected then v_i and v_k are also connected.

If for every pair of vertices $v, w \in V$, there is at least one path between v and w then we say the graph is **connected**. And if the graph is not connected then we say it is **disconnected**.

If a graph is disconnected then each of the maximal sets of connected vertices are called **connected components** or simply components. The graph depicted in Figure 5 has 8 vertices and 3 connected components. We can visually find these connected components by finding a pair of connected vertices, such as 2 and 6, and then repeatedly adding any vertices connected to the current set of connected vertices. Since connectedness is a transitive property, when we build up the connected component in our earlier algorithm it doesn't matter the order in which we consider the vertices to include. For instance, in the below example, we can start with vertices 2 and 6, and then add vertex 3 (which is connected to both 2 and 6 – even though the path between 3 and 6 goes through vertex 5 the pair 3 and 6 are still a connected pair of vertices), and finally add vertex 5 (which is connected to vertices 2,3, and 6); or we can start say with vertices 3 and 5, and then add vertex 6, and finally vertex 2 – we always end up with the same connected component since connectedness is transitive and the connected component is a maximal set (maximal roughly means keep adding as long as you can).

Note, we consider a vertex to be connected to itself by a path of length 0. Thus, an isolated vertex, such as vertex 8 in the below figure, is considered a connected component of size 1, and therefore, every vertex is in a connected component.

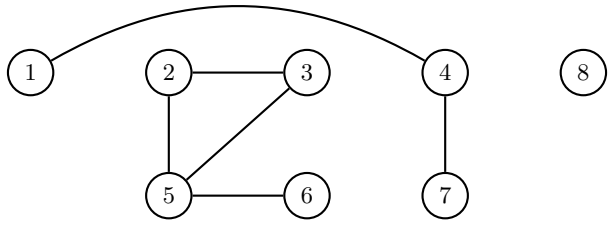

Figure 5: A graph with 3 connected components $\{1, 4, 7\}, \{8\}, \{3, 6, 5, 2\}$.

The minimal connected graph is a **tree**. Formally, **a tree is a connected, acyclic graph** where acyclic means there are no cycles.

There is one graph which is a tree on 3 vertices, but there are 2 trees for 4 vertices, and the number of trees grows quickly as n increases.

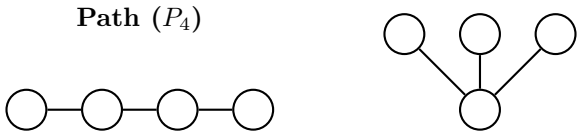

Figure 6: Possible trees on 4 vertices: the path graph P_4, and the star graph, denoted as $K_{1,3}$.

Key properties of trees on n vertices:

1. A tree has $n - 1$ edges.

2. Between every pair of vertices there is exactly one path.

We will discuss the first property in a later section. For the second property, consider a tree $T = (V, E)$. For every pair of vertices $v, w \in V$, there is at least one path between v and w in T since T is connected, and there is at most one path between v and w because otherwise we would have a cycle. Hence, there is exactly one path between every pair of vertices.

A forest is a generalization of a tree. Formally, a forest is a graph with no cycles. Therefore, a connected forest is a tree.

Consider a tree $T = (V, E)$. For an edge $e = \{v, w\} \in E$, if we remove e from the tree T, then we are left with 2 connected components, one containing v and one containing w, and each of these 2 connected components is a tree. Similarly if we remove a vertex v and all edges incident to v, then we break the tree into $d(v)$ smaller trees, one for each neighbor of v.

4.8 Subgraphs and Spanning Trees

For a graph $G = (V, E)$, for a subset $S \subset E$, we say $H = (V, S)$ is a **subgraph** of G. If the subgraph H is a tree then we say H is a spanning tree of G, though we will often simply refer to H as a tree in G.

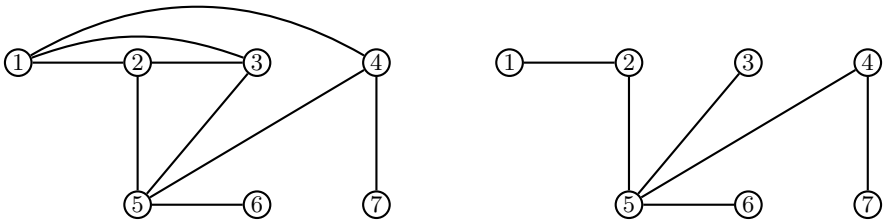

Figure 7: Original graph G shown next to one of its spanning trees T.

4.9 Bipartite Graphs

A graph $G = (V, E)$ is **bipartite** if there are two sets $L, R \subset V$ where $V = L \cup R$ and $L \cap R = \emptyset$, and for all edges $\{v, w\} \in E$ then either: $(v \in L, w \in R)$ or $(v \in R, w \in L)$. Two examples of bipartite graphs are shown in Figure 8. The analog of a complete graph is the complete bipartite graph, denoted as $K_{n,n}$ for $n + n$ vertices.

4.10 Some Proofs about Trees

We will now prove some basic properties about trees. The main result is that a tree on n vertices has exactly $n - 1$ edges. Before proving this result we need additional structural properties of trees.

A **leaf** is a vertex of degree one. The following lemma implies that every tree has at least one leaf.

Lemma 4.4. *For any $G = (V, E)$,*

$$\forall v \in V, d(v) \geq 2 \implies G \text{ has a cycle.}$$

Proof. We will do a proof by contradiction. Suppose every vertex has degree ≥ 2 but G is acyclic, i.e., G has no cycles. Consider a path $P = v_0, v_1, \ldots, v_\ell$ which is of maximum length. Since P is a path every vertex appears at most once in P. For the vertex v_ℓ, we know $\{v_{\ell-1}, v_\ell\} \in E$ since it's the edge on the path P,

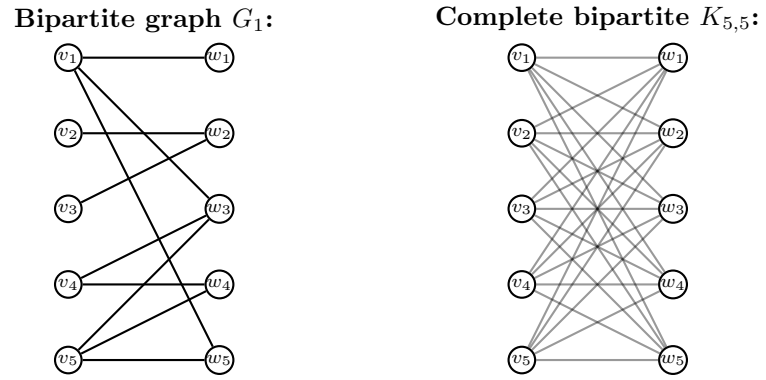

Figure 8: An example bipartite graph G_1, and the complete bipartite graph $K_{5,5}$.

and since $d(v_\ell) \geq 2$, then there is an additional edge $e^* = \{v_\ell, y\} \in E$ for some $y \in V$. If y appears on the path P, then $P \cup e^*$ has a cycle (containing y), which contradicts our assumption that G is acyclic. And if y does not appear on P, then there is a path P' by taking P and then appending e^*, and this new path P' is longer than P, which contradicts our assumption that P is a path of maximum length. Thus, in either case we have our contradiction, which completes the proof of the lemma. □

The contrapositive of Lemma 4.4 is that if a graph is acyclic, such as a tree, then there is at least one vertex of degree at most 1. Moreover, if the graph is a tree then we can conclude that there is a vertex with degree exactly 1, which is called a leaf vertex as mentioned before.

Corollary 4.5. *Let $G = (V, E)$ be an undirected graph.*

For any tree $T \subseteq E$, T has at least one leaf vertex.

Proof. A tree is connected and thus every vertex $v \in V$ has degree $d(v) \geq 1$; this is because a vertex v with $d(v) = 0$ is an isolated vertex which is disconnected from the rest of the graph. Since a tree is acyclic then by Lemma 4.4, there is a vertex v with $d(v) \leq 1$, and hence it must have degree $d(v) = 1$, in which case it is a leaf vertex by definition. □

We can now prove that a tree on n vertices has exactly $n - 1$ edges.

Lemma 4.6. *Let $G = (V, E)$ be an undirected graph where $n = |V|$. For any tree $T \subseteq E$,*

T has exactly $n - 1$ edges.

This can be proved using induction on n, the number of vertices – you will do so on your upcoming homework. To do standard induction we utilize Corollary 4.5 which shows that there is at least one leaf vertex. The lemma can also be proved without using Corollary 4.5 but then strong induction is needed as the smaller instances may have $n' < n - 1$ vertices.