

11 Graph Theory

This is from Chapter 11 of the text.

11.1 Introduction to Graph Theory

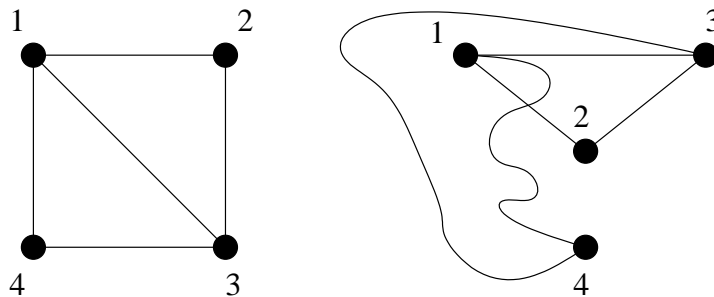
Definition 11.1. A graph $G = (V, E)$ consists of a non-empty set V of *vertices* and a set E of 2-element subsets of V , called *edges*.

For example

$$G = (\{1, 2, 3, 4\}, \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\}, \{1, 3\}\})$$

is a graph. It has four vertices and five edges. As this mess of brackets gets troublesome to write, we drop them when it doesn't confuse things and write an edge $\{x, y\}$ as xy .

The graph G can be represented pictorially as



We often write $V(G)$ for V and $E(G)$ for E if we don't define these sets for a graph G explicitly.

There are a lot of definitions related to the fact that $e = uv$ is an edge of G . We say

- the vertices u and v are *adjacent*,
- u and v are the *endpoints* of e ,
- u (and v) is *incident* with e ,

One can generalise the idea of a graph to

- *digraphs* by making edges ordered sets: uv and vu are different,
- *multigraphs* by making the edgeset E a multiset,
- *hypergraphs* by allowing edges to contain more than two vertices,
- *non-simple graphs* by allowing edges of the form uu .

We mostly don't consider these generalisations.

Definition 11.2. Let x and y be vertices in a graph G . An xy -walk is a finite alternating sequence of vertices and edges:

$$x = x_0, e_1, x_1, e_2, e_2, \dots, x_{n-1}, e_n, x_n = y$$

where $e_i = x_{i-1}x_i$ for $i = 1, \dots, n$.

Usually we just write the vertices

$$x_0x_1 \dots x_n,$$

or, if x and y are known, the edges

$$e_1e_2 \dots e_n.$$

The *length* of a walk is n , the number of edges in it. If $x = y$, the walk is called *closed*. A walk in which no edge is repeated is called a *trail*. A closed trail is a *circuit*. A walk in which no vertex is repeated is called a *path*. A closed walk in which only the endpoint is repeated is called a *cycle*.

Theorem 11.3. *If a graph has an xy -trail, then it has an xy -path.*

Intuitively, we just start with an xy -trail and 'chop' out the repeated bits until it is a path. A lot of times it is very hard to write the intuitive proof rigorously. Try it before you proceed. A nice way to prove this theorem is the following proof by contradiction, using a minimum counterexample.

Proof. By assumption, there exists an xy trail in G . Let $x = x_0, x_1, \dots, x_d = y$ be the shortest xy trail.

Towards contradiction, assume that the trail isn't a path. Then there exist some $0 \leq i < j \leq d$ such that $x_i = x_j$. But then

$$x_0, x_1, \dots, x_{i-1}, x_i, x_{j+1}, \dots, x_d$$

is an xy trail that is shorter than the original one. This is a contradiction, so this path is a trail. \square

Some common graphs have special names and notation:

- The complete graph on n vertices, K_n , has $V(K_n) = \{v_1, v_2, \dots, v_n\}$ and $E(K_n) = \{v_iv_j \mid 1 \leq i < j \leq n\}$.
- The n -cycle, C_n , has $V(C_n) = \{v_1, v_2, \dots, v_n\}$ and $E(C_n) = \{v_iv_{i+1} \mid i = 1, \dots, n-1\} \cup \{v_nv_1\}$.
- The n -path, P_n , has $V(P_n) = \{v_0, v_1, \dots, v_n\}$ and $E(P_n) = \{v_iv_{i+1} \mid i = 0, \dots, n-1\}$.
- The complete bipartite graph, $K_{m,n}$, has $V(K_{m,n}) = \{a_1, \dots, a_m\} \cup \{b_1, \dots, b_n\}$ and $E(K_{m,n}) = \{a_ib_j \mid i \in [m], j \in [n]\}$.
- The n -cube, Q_n , $V(Q_n)$ is the set of binary strings of length n , vertices u and v are adjacent if they differ in exactly one coordinate.

11.2 Subgraphs and Isomorphisms

Given a graph G , another graph G' is a *subgraph* of G , written $G' \leq G$ if $V(G') \subset V(G)$ and $E(G') \subset E(G)$.

A subgraph G' of G is *spanning* if $V(G') = V(G)$, or *induced* if $E(G') = \{uv \in E(G) \mid u, v \in V(G')\}$. A subgraph G' of G is a *proper* subgraph of G if $E(G')$ is a proper subset of $E(G)$.

Definition 11.4. A graph G is called *connected* if there is an xy -path for every two vertices $x, y \in V(G)$.

Any *maximal* connected subgraph of G , that is, a connected subgraph that is not a proper subgraph of any other subgraph of G , is called a component of G .

Remark 11.5. The word 'maximal' is different from 'maximum' and is not really used outside of mathematics. Both terms are defined with respect to a *partial ordering*. (See Section 7.3 of the text for more about partial orders.)

For graphs there are several different partial orders (or actually pre-orders) that we can consider.

Generally, when we say 'maximal' we are talking about the partial order 'subset inclusion' so a graph is maximal in a set of graphs if it is not a subgraph of any graph in the family but itself. When we say 'maximum' we usually mean with respect to the partial order 'number of vertices' or 'number of edges'. A finite set of graphs need not have a maximum with respect to subgraph inclusion.

Example 11.6. Let G be the graph containing a K_4 on the vertices $\{1, 2, 3, 4\}$ and a P_2 on the vertices $\{a, b, c\}$. G has two components, K_4 and P_2 . It has no maximum component if 'maximum' is taken with respect to the 'subgraph' relation, but we would usually say K_4 is the 'biggest', as it is maximum with respect to 'number of edges' and 'number of vertices'.

Definition 11.7. The *complement* \overline{G} of a graph G is the graph defined by $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \overline{E(G)}$, (where the universe used for complementation is the set $\{uv \mid u \neq v \in V(G)\}$ of all possible edges on $V(G)$).

Problem 11.1. Show that a subgraph G' of G is induced if and only if $\overline{G'}$ is a subgraph of \overline{G} .

Sol:

Let $G' \subset G$ be induced. Clearly $V(\overline{G'}) \subset V(\overline{G})$, so we have to show that $E(\overline{G'}) \subset E(\overline{G})$. Let $e \in E(\overline{G'})$ then $e \notin E(G')$ so $e \notin E(G)$. So $e \in E(\overline{G})$.

On the other hand, let $\overline{G'} \leq \overline{G}$. We want to show that G' is induced. For $u, v \in V(G')$, with $uv \in E(G)$ we want to show that $uv \in E(G')$. Well $uv \in E(G)$ implies that $uv \notin E(\overline{G})$ so not in $E(\overline{G'})$, so because $u, v \in V(G')$, $uv \in E(G')$.

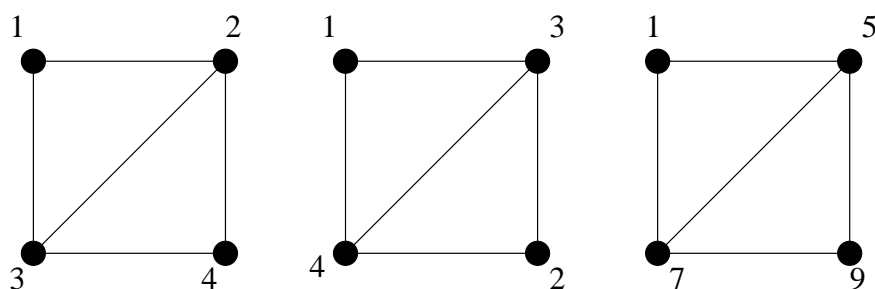
Often we don't care about the names of the vertices, and want to consider graphs to be the same if they look the same when ignoring these vertex names. For this we make the following definition.

Definition 11.8. Two graphs G and G' are *isomorphic* if there is a one-to-one function $f : V(G) \rightarrow V(G')$ such that

$$uv \in E(G) \iff f(u)f(v) \in E(G').$$

The function f is called an *isomorphism*.

Example 11.9. The following graphs are all isomorphic.

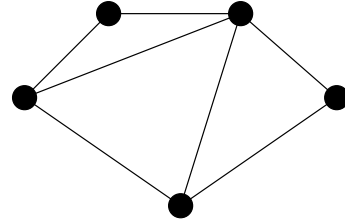
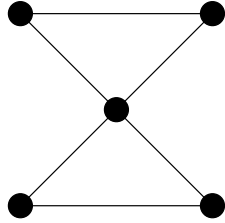


We usually consider graphs to be the same if they are isomorphic. We often want to decide if two graphs are isomorphic. Deciding if two graphs are isomorphic is usually pretty hard (this statement can be made rigorous). But if we are lucky, there are easy ways to prove that they are not.

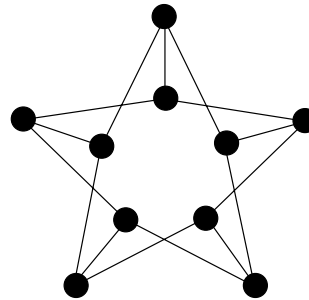
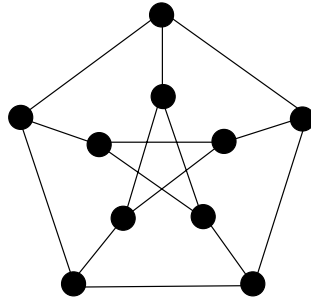
For example, if two graphs have a different number of vertices, or a different number of edges, or a different number of components, they are not isomorphic. Such properties, which are 'preserved by isomorphism', are called graph properties. Let's look at a couple more.

Definition 11.10. For a vertex v in a graph G , the *degree* of v , written $\deg(v)$, is the number of edges in G incident to v . The degree sequence of a graph G is the multiset of the degrees of the vertices of G . The minimum degree, $\delta(G)$, and maximum degree, $\Delta(G)$, of a graph G are the minimum and maximum degrees respectively in its degree sequence.

Clearly, degree sequence, minimum degree, and maximum degrees are graph properties. So the following two graphs, which have the same number of vertices, edges, and components, are not isomorphic as they have different degree sequences.



Problem 11.2. Are the following two graphs, with the same degree sequences, isomorphic?



Problem 11.3. 1. How many subgraphs are there of K_4 ?

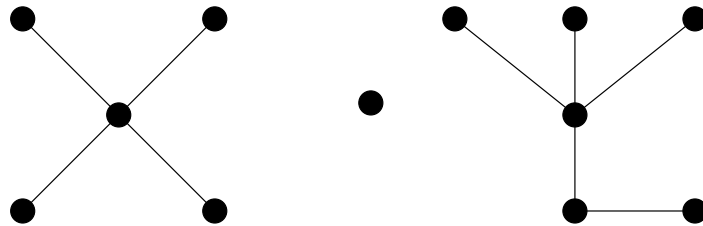
2. How many proper subgraphs?
3. How many induced subgraphs?
4. How many spanning subgraphs?
5. How many non-isomorphic subgraphs?

12 More Graph Theory

12.1 Trees

Definition 12.1. A *forest* is a graph with no cycles as subgraphs. A *tree* is a connected forest.

The following graph is a forest, each of its three components is a tree.



Definition 12.2. A *spanning tree* of a graph G is a spanning subgraph which is a tree.

Theorem 12.3. A graph G is connected if and only if it has a spanning tree.

Proof. Let G be a graph with a spanning tree T . Let x and y be any two vertices in G . As T is spanning, they are also in T . As T is connected, there is an xy -path $P = e_1e_2 \dots e_d$ in T . But $E(T) \subset E(G)$ so P is in G as well, that is, it is an xy -path in G . This way for any $x, y \in V(G)$, so G is connected.

On the other hand assume that G is connected. Let H be a minimal connected spanning subgraph of G , that is, a connected spanning subgraph, which itself has no connected spanning proper subgraph. We show by contradiction that H is a tree. Assume that it isn't. Then it contains a cycle $v_1v_2 \dots v_d$. Let $H' = H \setminus v_1v_2$. If H' is disconnected, then there exist vertices x and y of H such that there are no xy -paths in H' . So all xy -paths in H use v_1v_2 , that is, they are of the form $x = x_0x_1 \dots x_i v_1 v_2 x_{i+1} \dots x_m = y$. But then $x_0x_1 \dots x_i v_1 v_n v_{n-1} \dots v_2 x_{i+1} \dots x_m$ is an xy -walk in H' , contradicting the fact that there is no xy -path in H' . Thus H' is a smaller connected spanning subgraph, contradicting the choice of H . So by contradiction H is a tree. \square

Definition 12.4. A *leaf* or *pendant edge* of a tree is a vertex is of degree one.

Lemma 12.5. Any tree has a leaf.

Proof. Let G be a graph with no leaves. We show that G has a cycle, so is not a tree. This gives the contrapositive of the Theorem. Let $P = v_1v_2 \dots v_n$ be a longest path in G . As v_1 has some neighbour other than v_2 and there is no longer path than P , v_1 has a neighbour v_d on P , so v_1v_2, \dots, v_dv_1 is a cycle in G . \square

Theorem 12.6. For a tree T , $|V(T)| = |E(T)| + 1$.

Proof. The proof is by induction on $|V(T)|$. Since the only tree on one vertex is K_1 and the theorem is true for K_1 we assume that $n \geq 2$, and that the theorem is true for graphs on less than n vertices.

As T is a tree, it has a leaf v . Let $T' = T \setminus v$, the graph we get from T by removing v .

Claim 12.7. T' is a tree

Proof of claim. To see that it is connected, observe that the only paths in T that contain v , contain v as an endpoint. So for any vertices $x, y \in V(T) \setminus \{v\}$, the xy -path connecting them in T is also in T' . As $T' \leq T$ it has no cycles. So T' is a tree. \diamond

Now $|V(T)| - 1 = |V(T')|$, which by induction is $|E(T')| + 1 = |E(T)|$. So $|V(T)| = |E(T)| + 1$ as needed. \square

Definition 12.8. A *cut edge* of a graph G is any edge whose removal increases the number of components of G .

Theorem 12.9. Let $G = (V, E)$ be a graph. Then the following are equivalent.

1. G is a tree.
2. G is connected, but every edge is a cut edge.
3. G has no cycles, and $|V| = |E| + 1$.
4. G is connected, and $|V| = |E| + 1$.
5. Any two vertices of G are connected by a unique path.

Proof. From the previous theorem we have that (i) implies (iii) and (iv). We show that (iii) \Rightarrow (i) and (iv) \Rightarrow (ii) \Rightarrow (i). This shows the equivalence of the first four statements. As an exercise, show that statement (v) is also equivalent.

(iii) \Rightarrow (i): Assume that G has no cycles, so each component of G is a tree. For each component C we thus have that $|V(C)| = |E(C)| + 1$. So $|V(G)| = |V(E)| + c$ where c is the number of components. By $|V| = |E| + 1$, there is only one component, so it is a tree.

(iv) \Rightarrow (ii): Assume that G is connected and has $|V| = |E| + 1$. If there is some edge e that is not a cut-edge, then $G' = G \setminus \{e\}$, the graph we get by removing e , is connected, and so contains a spanning tree T . Thus $|V| = |V(G')| = |V(T)| = |E(T)| + 1 \leq |E(G')| + 1 = |E| + 2$, but this is a contradiction.

(ii) \Rightarrow (i): Let G be connected and every edge of G be a cut-edge. As G is connected it has a spanning tree. Since every edge is a cut edge, no proper subgraph has a spanning tree, so G itself is a spanning tree. □