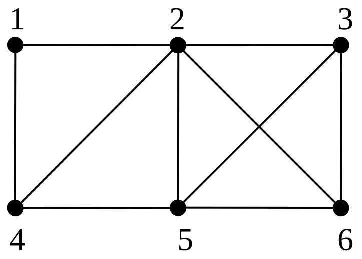
CMPS 101
Algorithms and Abstract Data Types
Graph Theory

Graphs

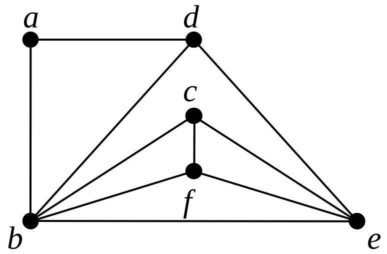
A *graph* G consists of an ordered pair of sets $G = (V, E)$ where $V \neq \emptyset$, and $E \subset V^{(2)} = \{\text{2-subsets of } V\}$. In other words E consists of *unordered* pairs of elements of V. We call $V = V(G)$ the *vertex* set, and $E = E(G)$ the *edge* set of G. In this handout, we consider only graphs in which both the vertex set and edge set are finite. An edge $\{x, y\}$, typically denoted *xy* or *yx*, is said to *join* its two *end vertices* x and y, and these ends are said to be *incident* with the edge *xy*. Two vertices are called *adjacent* if they are joined by an edge, and two edges are said to be *adjacent* if they have a common end vertex. A graph will usually be depicted as a collection of points in the plane (vertices), together with line segments (edges) joining the points.

Example 1 $V(G) = \{1, 2, 3, 4, 5, 6\}$, $E(G) = \{12, 14, 23, 24, 25, 26, 35, 36, 45, 56\}$



Two graphs G_1 and G_2 are said to me *isomorphic* if there exists a bijection $\phi: V(G_1) \rightarrow V(G_2)$ such that for any $x, y \in V(G_1)$, the pair *xy* is an edge of G_1 if and only if the pair $\phi(x)\phi(y)$ is an edge of G_2. In other words, the function ϕ must preserve all incidence relations amongst the vertices and edges in G_1 . We write $G_1 \cong G_2$ to mean that G_1 and G_2 are isomorphic.

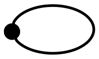
Example 2 Let G_1 be the graph from the previous example, and define G_2 by $V(G_2) = \{a, b, c, d, e, f\}$, $E(G_2) = \{ab, ad, bc, bd, be, bf, ce, cf, de, ef\}$. G_2 can be drawn as



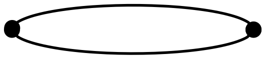
Define a mapping $\phi: V(G_1) \rightarrow V(G_2)$ by $1 \rightarrow a,\ 2 \rightarrow b,\ 3 \rightarrow c,\ 4 \rightarrow d,\ 5 \rightarrow e,\ 6 \rightarrow f.$ One checks easily that ϕ is an isomorphism.

Isomorphic graphs are indistinguishable as far as graph theory is concerned. In fact, graph theory can be defined to be the study of those properties of graphs that are preserved by isomorphisms. Thus a graph is not a picture, in spite of the way we visualize it. Instead, a graph is a combinatorial object consisting of two abstract sets, together with some incidence data relating those sets.

Notice that our definition of a graph does not allow for the existence of an edge joining a single vertex to itself (sometimes called a loop), since an edge $xy = \{x, y\}$ must be a 2-*element* subset of V, and therefore must have distinct ends.



Neither does our definition allow two distinct edges to join the same pair of vertices (parallel edges), since the edge set E, being a set of pairs of vertices, cannot contain the same pair twice.



When these types of edges are allowed, we call the resulting structure a *multi-graph*. Some authors use the term "graph" to denote what we have designated as a multi-graph. Those authors would call our notion of graph (i.e. a graph in which loops and parallel edges do not occur) a *simple graph*.

The *degree* of a vertex $x \in V(G)$, denoted $\deg(x)$, is the number of edges incident with x, or equivalently, the number of vertices adjacent to x. Referring to Example 1 above, we see that $\deg(1) = 2$, $\deg(2) = 5$ and $\deg(6) = 3$. The *degree sequence* of a graph consists of its vertex degrees arranged in increasing order. The graph in Example 1 has degree sequence $(2, 3, 3, 3, 4, 5)$. Observe that the graph in Example 2 has the same degree sequence. Clearly if $\phi: V(G_1) \to V(G_2)$ is an isomorphism, then $\deg\big(\phi(x)\big) = \deg(x)$ for any $x \in V(G)$, and hence isomorphic graphs must have the same degree sequence. This is our first example of an *isomorphism invariant*, a graph property that is preserved by isomorphisms.

Observe that

$$\sum_{x \in V(G)} \deg(x) = 2|E(G)|$$

since each edge, having two distinct ends, contributes 2 to the sum on the left. This is sometimes known as the *Handshake Lemma* for it says that the number of hands shaken at a party is exactly twice the number of handshakes.

Exercise 1 Show that the number vertices of odd degree in any graph must be even. (Hint: suppose G contains an odd number of odd vertices. Argue that the left hand side of the above equation is then odd, while the right hand side is clearly even.)

Given any $x, y \in V(G)$, an x-y *walk* in G, is a sequence of vertices $x = v_0, v_1, v_2, \ldots, v_{k-1}, v_k = y$ for which $v_{i-1}v_i \in E(G)$ for $1 \leq i \leq k$. We call x the *origin* and y the *terminus* of the walk. Note these need not be distinct. If $x = y$, the walk is said to be *closed*. The *length* of the walk is k, the number of edge traversals performed in going from x to y along the sequence. Since the edges of a graph have no inherent direction, we do not distinguish between the above sequence and its reversal: $y = v_k, v_{k-1}, \ldots, v_2, v_1, v_0 = x$. Thus the designation as to which vertex in a walk is the origin and which is the terminus is arbitrary. If G contains an x-y walk, we say that y is *reachable* from x (equivalently x is *reachable* from y). A special case of a closed walk is the *trivial walk* $x = v_0 = y$ of length 0, in which *no* edges are traversed.

A walk in which no edge is traversed more than once is called a *trail*, and a trail in which no vertex is visited more than once (except possibly when origin = terminus) is called a *path*. Observe that the trivial walk is both a trail and a path. A non-trivial closed path is called a *cycle*. Since our graphs have no loops or parallel edges, the minimum possible length for a cycle is 3.

Example 3 Referring again to Example 1 we have:

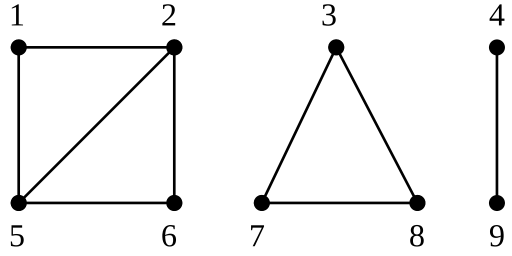
- a cycle of length 3: 2, 5, 6, 2
- a cycle of length 6: 1, 2, 3, 6, 5, 4, 1
- a 1-6 path of length 5: 1, 4, 2, 5, 3, 6
- a 1-6 path of length 2: 1, 2, 6
- a 3-1 trail that is not a path: 3, 2, 5, 6, 2, 1
- a 3-1 walk that is not a trail: 3, 5, 2, 4, 5, 2, 1
- the trivial 1-1 path: 1 (though this is a closed path, it is not a cycle)

Note that the reachability relation could just as well have been defined in terms of paths instead of walks since, if *G* contains an *x*-*y* walk that is not a path, we can, by eliminating some cycles, replace it by an *x*-*y* path. For instance the 3-1 walk in Example 1 above (3, 5, 2, 4, 5, 2, 1) contains the cycle (5, 2, 4, 5), and can be replaced by the 3-1 path (3, 5, 2, 1).

The *distance* from x to y , denoted $\delta(x, y)$, is defined as follows. If y is reachable from x, then $\delta(x, y)$ is the length of a shortest *x*-*y* path. If y is not reachable from x (i.e. if no *x*-*y* path exists), then $\delta(x, y)$ is infinity. The Single Source Shortest Path (SSSP) problem is this: given a distinguished vertex $s \in V(G)$ called the *source*, determine $\delta(s, x)$ for all $x \in V(G)$, and for each x that is reachable from s, determine a shortest *s*-*x* path in *G*. We will learn two efficient algorithms that solve this problem.

A graph *G* is said to be *connected* iff y is reachable from x for every pair of vertices $x, y \in V(G)$. If *G* is not connected, it is called *disconnected*. Examples 1 and 2 above are clearly connected, while the following graph is disconnected.

Example 4 $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $E = \{12, 15, 25, 26, 56, 37, 38, 78, 49\}$

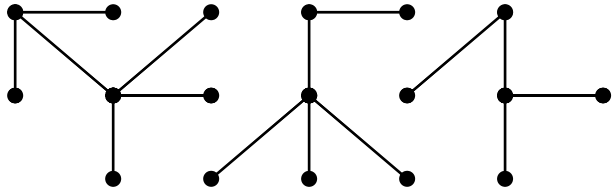


A *subgraph* of a graph *G* is a graph *H* in which $V(H) \subseteq V(G)$, and $E(H) \subseteq E(G)$. In the above example, $(\{1, 2, 5\},\ \{12, 15, 25\})$ is a connected subgraph, while $(\{2, 3, 6, 7\},\ \{26, 37\})$ is a disconnected subgraph. A subgraph *H* is called a *connected component* of *G* if it is (i) connected, and (ii) maximal with respect to property (i), i.e. any other subgraph of *G* that properly contains *H* is disconnected. Observe Example 4 has three connected components: $(\{1, 2, 5, 6\}, \{12, 15, 25, 26, 56\})$, $(\{3, 7, 8\}, \{37, 38, 78\})$ and $(\{4, 9\}, \{49\})$. Obviously a graph is connected if and only if it has exactly one connected component.

Trees

A graph is called *acyclic* (or a *forest*) if it contains no cycles. A *tree* is a graph that is both connected and acyclic. The connected components of an acyclic graph are obviously trees. The following example is a forest with three connected components.

Example 5



# of vertices:	8	6	5
# of edges:	7	5	4

Observe that in each tree of this forest, the number of edges is one less that the number of vertices. This fact holds in general for all trees. The following lemmas demonstrate how the independent properties of connectedness and acyclicity are related.

Lemma 1 If T is a tree with n vertices and m edges, then $m = n - 1$.
Proof:
This result was proved in the handout on Induction Proofs by induction on n. We prove it here by induction on m. If $m = 0$ then T can have only one vertex, since T is connected. Thus $n = 1$, and $m = n - 1$, establishing the base case. Now let $m > 0$ and assume that any tree T' with fewer than m edges satisfies $|E(T')| = |V(T')| - 1$. Pick an edge $e \in E(T)$ and remove it. The resulting graph consists of two trees T_1, T_2, each having fewer than m edges. Suppose T_i has m_i edges and n_i vertices ($i = 1, 2$). Then the induction hypothesis gives $m_i = n_i - 1$ ($i = 1, 2$). Also $n = n_1 + n_2$ since no vertices were removed. Therefore $m = m_1 + m_2 + 1 = (n_1 - 1) + (n_2 - 1) + 1 = n_1 + n_2 - 1 = n - 1$, as required. ■

Lemma 2 If G is an acyclic graph with n vertices, m edges, and k connected components, then $m = n - k$.
Proof:
Let the connected components of G (which are necessarily trees) be denoted $T_1, T_2, \ldots, T_k$. Suppose T_i has m_i edges and n_i vertices respectively ($1 \le i \le k$). By Lemma 1 we have $m_i = n_i - 1$ ($1 \le i \le k$). Therefore

$$m = \sum_{i=1}^{k} m_i = \sum_{i=1}^{k} (n_i - 1) = \sum_{i=1}^{k} n_i - \sum_{i=1}^{k} 1 = n - k ,$$

as claimed. ///

Lemma 3 If G is a connected graph with n vertices and m edges, then $m \ge n - 1$.
Proof:
Our proof is a generalization of that of Lemma 1, again by induction on m. If $m = 0$, then G, being connected, can have only one vertex, hence $n = 1$. Therefore $m \ge n - 1$ reduces to $0 \ge 0$, showing that the base case is satisfied.

Let $m > 0$, and assume for any connected graph G' with fewer than m edges that $|E(G')| \ge |V(G')| - 1$. Remove an edge $e \in E(G)$ and let $G - e$ denote the resulting subgraph. We have two cases to consider.

<u>Case 1</u>: $G - e$ is connected. We note that $G - e$ has n vertices and $m - 1$ edges, so the induction hypothesis gives $m - 1 \ge n - 1$. Certainly then $m \ge n - 1$, as was claimed.

<u>Case 2</u>: $G-e$ is disconnected. In this case $G-e$ consists of two connected components. (**See the claim and proof below.) Call them H_1 and H_2, and observe that each component contains fewer than *m* edges. Suppose H_i has m_i edges and n_i vertices ($i=1, 2$). The induction hypothesis gives $m_i \geq n_i - 1$ ($i=1, 2$). Also $n = n_1 + n_2$ since no vertices were removed. Therefore

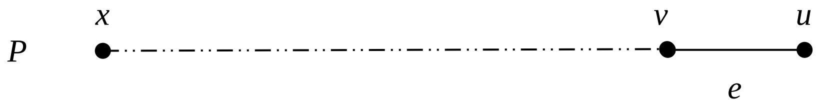
$$m \;=\; m_1 + m_2 + 1 \;\geq\; (n_1 - 1) + (n_2 - 1) + 1 \;=\; n_1 + n_2 - 1 \;=\; n - 1,$$

and therefore $m \geq n-1$ as required. ///

Claim**: Let *G* be a connected graph and $e \in E(G)$, and suppose that $G-e$ is disconnected. (Such an edge *e* is called a *bridge*). Then $G-e$ has exactly two connected components.

Proof:

Since $G-e$ is disconnected, it has at least two components. We must show that it also has at most two components. Let *e* have end vertices *u*, and *v*. Let C_u and C_v be the connected components of $G-e$ that contain *u* and *v* respectively. Choose $x \in V(G)$ arbitrarily, and let *P* be an *x-u* path in *G* (note *P* exists since *G* is connected.) Either *P* includes the edge *e*, or it does not. If *P* does not contain *e*, then *P* remains intact after the removal of *e*, and hence *P* is an *x-u* path in $G-e$, whence $x \in C_u$. If on the other hand *P* does contain the edge *e*, then *e* must be the last edge along *P* from *x* to *u*.



In this case $P-e$ is an *x-v* path in $G-e$, whence $x \in C_v$. Since *x* was arbitrary, every vertex in $G-e$ belongs to either C_u or C_v, and therefore $G-e$ has at most two connected components. ///

Lemma 4 If *G* is a graph with *n* vertices, *m* edges, and *k* connected components, then $m \geq n-k$.

Proof:

Let $H_1, H_2, \ldots, H_k$, be the connected components of *G*. Let n_i and m_i denote the number of vertices and edges, respectively, of H_i, for $1 \leq i \leq k$. By Lemma 3 we have $m_i \geq n_i - 1$, for $1 \leq i \leq k$, and therefore

$$m \;=\; \sum_{i=1}^{k} m_i \;\geq\; \sum_{i=1}^{k} (n_i - 1) \;=\; \sum_{i=1}^{k} n_i - \sum_{i=1}^{k} 1 \;=\; n - k\,.$$

whence $m \geq n-k$ as claimed. ///

Lemma 5 Let *G* be a connected graph with *n* vertices and *m* edges. Suppose also that $m = n-1$. Then *G* is acyclic, and hence a tree.

Proof:

Suppose *G* is connected and $m = n-1$. Assume, to get a contradiction, that *G* is not acyclic. Let *e* be any edge belonging to any cycle in *G*. Remove *e* from *G*, and denote the resultant graph by $G-e$. Observe that $G-e$ has $m-1$ edges and *n* vertices, respectively. Since *e* is a cycle edge, its removal does not disconnect *G*, and therefore $G-e$ is also connected. Lemma 3 above then gives $m-1 \geq n-1$, whence $m \geq n$. But then $m = n-1$ gives $n-1 \geq n$, a contradiction. Therefore our original assumption was false, and therefore *G* is acyclic, as claimed. Being connected, *G* is also a tree. ///

Lemma 6 Let G be an acyclic graph with n vertices and m edges. Suppose also that $m = n - 1$. Then G is connected, and hence a tree.

Proof:

Suppose G is acyclic and $m = n - 1$. Let k be the number of connected components of G. By Lemma 2 we have $m = n - k$, whence $n - 1 = n - k$, and hence $k = 1$, showing that G is connected, as claimed. ///

Lemma 7 Let G be a connected graph with n vertices and m edges. Suppose also that $m = n$. Then G contains exactly one cycle. (Such a graph is called unicyclic.)

Proof:

G contains at least one cycle since otherwise G is a tree, and hence $m = n - 1$ (by Lemma 1), contrary to hypothesis. If G contained two distinct cycles, say C_1 and C_2, we could find edges $e_1 \in E(C_1) - E(C_2)$ and $e_2 \in E(C_2) - E(C_1)$. Removing these two edges gives a connected graph $H = G - e_1 - e_2$ with $|V(H)| = n$ and $|E(H)| = n - 2$, contradicting Lemma 3. ///

Consider the following three properties of a graph $G = (V, E)$ in light of Lemmas 1, 5, and 6:

(i) G is connected,
(ii) G is acyclic
(iii) $|E| = |V| - 1$.

We see that these properties are logically dependent in the sense that if any two hold, then the third must also hold. Lemma 1 states that (i) and (ii) together imply (iii), Lemma 5 says that (i) and (iii) imply (ii), and Lemma 6 says (ii) and (iii) imply (i). The following theorem summarizes these and other facts about trees.

Theorem 1 (The Treeness Theorem) Let $G = (V, E)$ be a graph with $|V| = n$ and $|E| = m$. Then the following statements are equivalent.

a) G is a tree (i.e. connected and acyclic).
b) G contains a *unique* x-y path for any $x, y \in V$.
c) G is connected, but if any edge is removed, the resulting graph $G - e$ is disconnected.
d) G is connected, and $m = n - 1$.
e) G is acyclic, and $m = n - 1$.
f) G is acyclic, but if any edge is added (joining two non-adjacent vertices), then the resulting graph $G + e$ contains a *unique* cycle.

Note: This is theorem B.2 in Cormen, Leiserson, Rivest, & Stein (p.1085 in 2nd ed., p.1174 in 3rd ed.)

Proof: As mentioned in the preceding paragraph, Lemmas 1, 5, and 6 have already established the equivalences $(a)\Leftrightarrow(d)\Leftrightarrow(e)$.

$(a)\Rightarrow(b)$: Suppose G is a tree and let $x, y \in V(G)$. If $x = y$, then the trivial path is the only x-y path in G, since being a tree, G has no cycles. Suppose $x \neq y$. Since G is connected, there exists at least one x-y path in G. Assume, to get a contradiction, that G contains two distinct x-y paths. Call them p_1 and p_2. By traveling along p_1 from x to y, then along p_2 from y to x, we obtain a closed walk in G that begins and ends at x. If no vertex (other than x) is repeated in this walk, then we have found a cycle in G. If some vertex is repeated in this walk, we can obtain a cycle as follows. Travel along p_1 from x to the first repeated vertex, then back to x along p_2. Therefore G contains a cycle, contradicting that it is a tree. This contradiction shows our assumption was false, and hence two different x-y paths cannot exist in G, which therefore contains a unique x-y path.

$(b) \Rightarrow (c)$: Suppose G contains a unique x-y path for any $x, y \in G$. Then G is certainly connected. Pick any edge $e = \{x, y\} \in E$ and remove it. Since e constitutes a path joining its two ends (x and y) and since there are no other such paths by hypothesis, the resulting graph $G - e$ contains no x-y path, and is therefore disconnected.

$(c) \Rightarrow (a)$: Suppose G is connected, and if any edge is removed from G, the resulting graph is disconnected. Assume, to get a contradiction, that G contains a cycle. Then the removal of any edge on that cycle would not disconnect G, contrary to hypothesis. It follows that no such cycle can exist, and hence G is acyclic. Since G is also connected, it is a tree.

$(a) \Rightarrow (f)$: Suppose G is a tree (i.e. connected and acyclic). By Lemma 1 we also have $m = n - 1$. Pick any two non-adjacent vertices and join them with a new edge e. The resulting graph $G + e$ is still connected, and $|E(G + e)| = m + 1 = n = |V(G + e)|$. Lemma 7 now says that $G + e$ contains exactly one cycle.

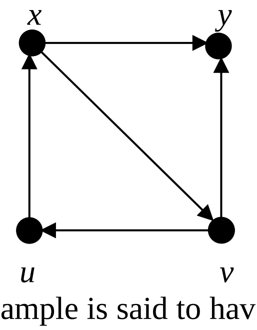
$(f) \Rightarrow (a)$: Suppose G is acyclic, and if a new edge e is inserted between any two non-adjacent vertices, then the resulting graph $G + e$ is unicyclic. Pick any $x, y \in V$ with $x \neq y$. If x and y are adjacent, then certainly y is reachable from x. If x and y are non-adjacent, insert a new edge e joining x to y, and call the resulting cycle C. Observe that the edges of $C - e$ constitute an x-y path in G, so in this case also, y is reachable from x. Since x and y were arbitrary, we conclude that G is connected, and therefore a tree.

This completes the proof of the Treeness Theorem. ■

Directed Graphs

A *Directed Graph* (or *Digraph*) $G = (V, E)$ is a pair of sets, where the vertex set $V = V(G)$ is, as before, finite and non-empty, and the edge set $E = E(G) \subseteq V \times V$, i.e. E consists of *ordered* pairs of vertices.

Example 6 $V = \{x, y, u, v\}$ and $E = \{(x, y), (u, x), (v, y), (v, u), (x, v)\}$



The directed edge (x, y) in the above example is said to have *origin* x and *terminus* y, and we say that x is *adjacent* to y. The origin and terminus of a directed edge are said to be *incident* with that edge. Two edges are called *adjacent* if they have a common end vertex, so for instance (x, y) in the above example is adjacent to (u, x). The *in degree* of a vertex is the number of edges having that vertex as terminus, and it's *out degree* is the number of edges having that vertex as origin. The *degree* of a vertex is the sum if it's in degree and out degree. Thus in the above example $\mathrm{id}(x)=1$, $\mathrm{od}(x)=2$, and $\deg(x)=3$. The analog of the handshake lemma for directed graphs is

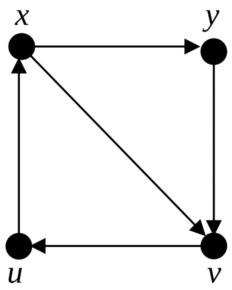
$$\sum_{x\in V(G)}\mathrm{id}(x) \;=\; \sum_{x\in V(G)}\mathrm{od}(x) = \;|E(G)|$$

As in the undirected case, there is a simple notion of isomorphism for directed graphs. Two digraphs G_1 and G_2 are said to me *isomorphic* if there exists a bijection $\phi: V(G_1)\to V(G_2)$ such that for any $x, y\in V(G_1)$, the ordered pair (x, y) is a directed edge of G_1 if and only if the ordered pair $(\phi(x), \phi(y))$ is a directed edge of G_2. Thus ϕ preserves incidence relations and directionality amongst the vertices and edges of G_1. We write $G_1\cong G_2$ to mean that G_1 and G_2 are isomorphic.

A *directed path P* in a digraph is a finite sequence of vertices P: $v_0, v_1, v_2, \ldots, v_{k-1}, v_k$ such that $(v_{i-1}, v_i)\in E$ for all $1\le i\le k$. As in the undirected case, we require that all vertices be distinct (except possibly v_0 and v_k), and that no edge be traversed more than once. If it so happens that the initial and terminal vertices are the same, $v_0 = v_1$, the path is called a *directed cycle*. The length of such a path is *k*, the number of edges traversed. If $x = v_0 \ne v_k = y$, we call *P* a directed *x*-*y* path. Notice that, unlike the undirected case, a directed *x*-*y* path and a directed *y*-*x* path are not the same thing. We say that $y\in V(G)$ is *reachable* from $x\in V(G)$ if *G* contains a directed *x*-*y* path. Observe that for digraphs, the reachability relation is reflexive (*x* is reachable from *x* via the trivial path with no edges), transitive (if *y* is reachable from *x*, and *z* is reachable from *y*, then *z* is reachable from *x*), but not symmetric (*y* may be reachable from *x* without *x* being reachable from *y*).

A digraph *G* is said to be *strongly connected* if for all $x, y\in V(G)$, both *x* is reachable from *y*, and *y* is reachable from *x*. Notice that the digraph in Example 6 above is not strongly connected, since for instance, *u* is not reachable from *y*. The following example *is* strongly connected.

Example 7 $V=\{x, y, u, v\}$ and $E=\{(x, y), (u, x), (y, v), (v, u), (x, v)\}$



More generally, a subset $S\subseteq V(G)$ is said to be *strongly connected* if for all $x, y\in S$, both *x* is reachable from *y*, and *y* is reachable from *x*. Furthermore, a subset $S\subseteq V(G)$ is said to be a *strongly connected component* of the digraph *G* if it is (i) strongly connected, and (ii) maximal with respect to property (i), i.e. any other subset of $V(G)$ that properly contains *S* is not strongly connected. Obviously *G* is strongly connected iff it has just one strongly connected component, namely $V(G)$ itself. Going back to the digraph in Example 6, we see that it has 2 strongly connected components: $\{x, u, v\}$ and $\{y\}$.

If we replace each directed edge in a digraph G with an undirected edge, we obtain an (undirected) graph known as the *underlying undirected graph* of G. Note that two non-isomorphic digraphs, such as Examples 6 and 7 above, can have the very same underlying graph.

Representations of Graphs and Digraphs

We discuss three methods for representing graphs and digraphs in terms of standard data structures available in most computer languages. They are called the *Incidence Matrix*, the *Adjacency Matrix*, and the *Adjacency List* representations respectively. In what follows we suppose $G=(V,E)$ to be a graph (directed or undirected) with $|V|=n$ and $|E|=m$.

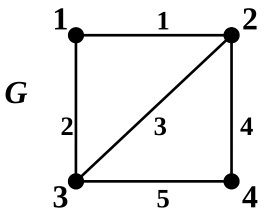
The *Incidence Matrix* $I(G)$ requires that both the vertex set $V(G)$ and the edge set $E(G)$ be ordered. For this purpose we suppose that $V=\{x_1,x_2,\ldots,x_n\}$ and $E=\{e_1,e_2,e_3,\ldots,e_m\}$. Then $I(G)$ is an $n\times m$ rectangular matrix. Row i corresponds to vertex x_i, for $1\le i\le n$. Column j corresponds to edge e_j ($1\le j\le m$), and contains zeros everywhere except for the two rows corresponding to the ends of e_j. If G is an undirected graph, these two rows contain 1s. If G is a directed graph, the row corresponding to the origin of e_j contains -1, while the row corresponding to the terminus of e_j contains $+1$. Thus $I(G)=(I_{i\,j})$ where in the undirected case:

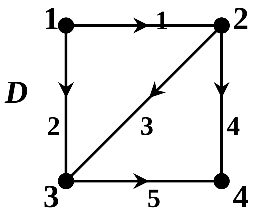
$$I_{i\,j}=\begin{cases}1 & \text{if } x_i \text{ is incident with } e_j\\ 0 & \text{otherwise}\end{cases}$$

and in the directed case:

$$I_{i\,j}=\begin{cases}1 & \text{if } x_i \text{ is the terminus of } e_j\\ -1 & \text{if } x_i \text{ is the origin of } e_j\\ 0 & \text{otherwise}\end{cases}$$

We illustrate on the graph G and digraph D pictured below.





$$I(G)=\begin{pmatrix}1&1&0&0&0\\1&0&1&1&0\\0&1&1&0&1\\0&0&0&1&1\end{pmatrix}\qquad I(D)=\begin{pmatrix}-1&-1&0&0&0\\1&0&-1&-1&0\\0&1&1&0&-1\\0&0&0&1&1\end{pmatrix}$$

The *Adjacency Matrix* $A(G)$ requires that only the vertex set come equipped with an order. It is a square matrix of size $n \times n$. Let $V = \{x_1, x_2, \ldots, x_n\}$ and define the i^{th} row and j^{th} column of $A(G)$ to be 1 if there is an edge from x_i to x_j, and 0 otherwise. Thus we have $A(G) = (A_{ij})$ where in the undirected case

$$A_{i\,j} = \begin{cases} 1 & \text{if } x_i \text{ is adjacent to } x_j \\ 0 & \text{otherwise} \end{cases}$$

and in the directed case

$$A_{i\,j} = \begin{cases} 1 & \text{if there is an edge with origin } x_i \text{ and terminus } x_j \\ 0 & \text{otherwise} \end{cases}$$

Observe that for an undirected graph $A = A(G)$ is a symmetric matrix (i.e. $A = A^T$, where A^T denotes the transpose of A.) The Adjacency Matrix for a directed graph is not in general symmetric. We illustrate on the same graph and digraph as before.

$$A(G) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \qquad A(D) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The *Adjacency List* representation of G consists of an array $\text{adj} = \text{adj}(G)$ of n lists. As above, let $V = \{x_1, x_2, \ldots, x_n\}$. Then in the undirected case, the array element $\text{adj}[i]$ is a list containing the vertices adjacent to x_i $(1 \le i \le n)$. In the directed case $\text{adj}[i]$ is a list containing the termini of edges having origin x_i.

<u>Undirected case:</u>

adj[1]: list of neighbors of x_1
adj[2]: list of neighbors of x_2
adj[3]: list of neighbors of x_3
$\vdots$
adj[i]: list of neighbors of x_i
$\vdots$
adj[n]: list of neighbors of x_n

<u>Directed Case:</u>

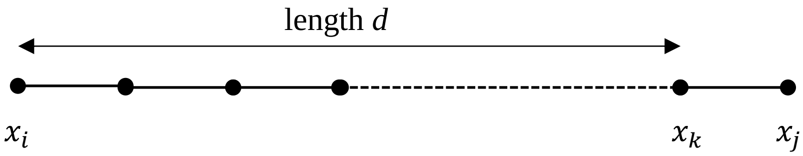
adj[1]: list of termini of edges having origin x_1
adj[2]: list of termini of edges having origin x_2
adj[3]: list of termini of edges having origin x_3
$\vdots$
adj[i]: list of termini of edges having origin x_i
$\vdots$
adj[n]: list of termini of edges having origin x_n

Again we illustrate on the same examples.

$$\text{adj}(G) = \begin{cases} 1: & 2 \quad 3 \\ 2: & 1 \quad 3 \quad 4 \\ 3: & 1 \quad 2 \quad 4 \\ 4: & 2 \quad 3 \end{cases} \qquad \text{adj}(D) = \begin{cases} 1: & 2 \quad 3 \\ 2: & 3 \quad 4 \\ 3: & 4 \\ 4: & \end{cases}$$

Observe that the adjacency list representation is nothing more than the sparse matrix representation (as in pa3) of the adjacency matrix.

Exercise
Let *G* be a graph, $A = A(G)$ its adjacency matrix, and $d \geq 0$. Show that the number of walks in *G* from x_i to x_j of length *d* is given by the ij^{th} entry of A^d. Hint: Use weak induction on *d* starting at $d = 0$, noting that $A^0 = I$, the identity matrix. For the induction step observe that any walk from x_i to x_j of length $d + 1$ consists of a walk from x_i to some intermediate vertex x_k of length *d*, followed by the traversal of a single edge from x_k to x_j.



The number of such walks from x_i to x_k is $(A^d)_{ik}$ by the induction hypothesis, and the number of such edges (0 or 1) is A_{kj}.

Exercise
State and prove a theorem, analogous to the one above, for directed graphs.

Exercise
The purpose of this exercise is to discover how the previous result can be used to solve the *all pairs shortest paths* (APSP) problem:

> Given a graph G with n vertices, $V(G) = \{x_1, x_2, \dots \dots, x_n\}$, do the following for all pairs i, j satisfying $1 \leq i \leq j \leq n$: (1) determine $d(x_i, x_j)$, and (2) if $d(x_i, x_j) < \infty$, determine a shortest path from x_i to x_j.

Fix such a pair i, j.

a. Show that if $d(x_i, x_j) < \infty$ (i.e. x_j is reachable from x_i), then $d(x_i, x_j) \leq n - 1$.
b. Show that a minimum length walk from x_i to x_j is necessarily an x_i-x_j path, and hence a shortest such path.
c. If *M* is an $n \times n$ matrix, let M_{ij} denote its ij^{th} entry, i.e. the element in its i^{th} row, j^{th} column. Suppose that $(A^k)_{ij}$ is the first non-zero term in the integer sequence: $I_{ij},\ A_{ij},\ (A^2)_{ij},\ (A^3)_{ij},\ \dots \dots,\ (A^{n-1})_{ij}$. Show that then $d(x_i, x_j) = k$.
d. (More difficult). Part (1) of APSP is solved by (c). Figure out how to solve part (2) of APSP.