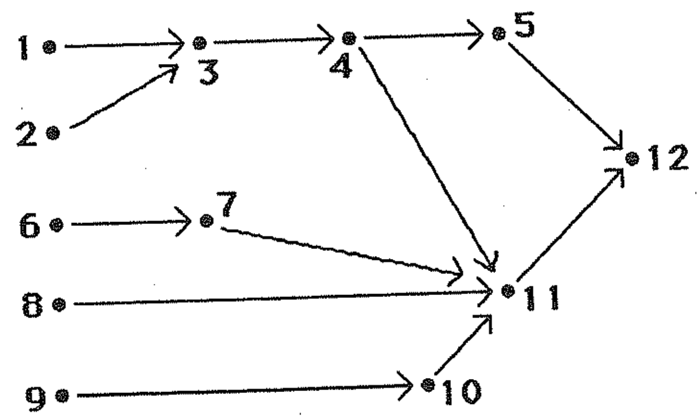
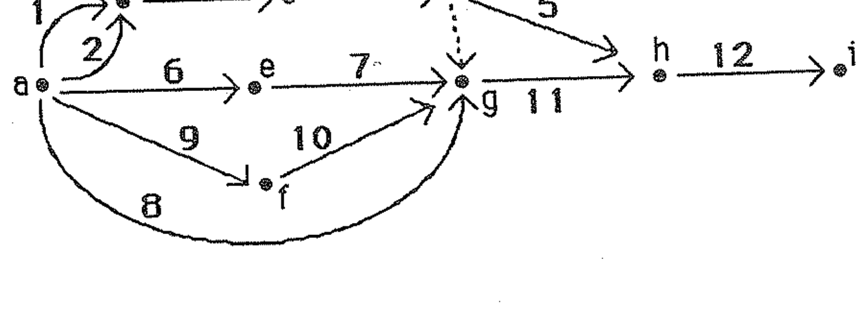


(b) If we want to let the activities be represented by edges, then we need to identify certain points in time to be the vertices. One such point is the starting time of the entire project (call it vertex a). Another is the point at which the history term paper is ready to write (b). To get from point a to point b, Sam needs to perform tasks 1 and 2 (but they can be done in either order). Thus we put two edges from a to b, namely 1 and 2. Similarly, we put edge 3 from vertex b to vertex c (the point at which the draft of the history paper has been written and is ready to edit), and edge 4 from c to d (the point at which the paper has been finished). We let vertex e be the beginning of the history exam and vertex f be the beginning of the discrete math exam, so we put edge 6 from a to e and edge 9 from a to f. Now let vertex g represent the beginning of the celebration; since this point must come after the two exams and the handing in of the discrete math exercise set, we put edges 7, 10, and 8 from vertices e, f, and a, respectively, to vertex g. Furthermore, point d must precede point g, since the history paper has to be finished. Therefore we put an edge from d to g. There is no activity on this edge; it just represents a precedence relationship. Next we put edges 5 and 11 from d and g (respectively) to a vertex h representing the point at which all the work at school has been done (including returning the word processor and celebrating). Finally, edge 12 (flying home) goes from this vertex to a final vertex, i, representing the completion of the entire project. The digraph is as shown here.



29. (a) This graph is not locally finite, since each vertex is adjacent to an infinite number of other vertices, namely all the points on the circle of radius 1 inch, centered at that vertex. An equilateral triangle 1 inch on a side shows that K_3 is a subgraph of this graph. Therefore K_n is subgraph for all $n \le 3$. On the other hand, it is clearly impossible to have four points in the plane all 1 inch from the others, so K_n is not a subgraph if $n \ge 4$. Finally, C_n is a subgraph for all $n \ge 3$, since the regular n-gon with sides of length 1 inch forms such a subgraph.

(b) This graph is not locally finite, since $\{1\}$ is adjacent to all the vertices $\{1, k\}$ for all $k \ge 2$. We claim that K_n is subgraph for all n. To see this, just note that $\{1\}$, $\{1, 2\}$, $\{1, 2, 3\}$, ... are all adjacent. In fact even more is true than that K_n is a subgraph for all n; we can say that K_∞ is a subgraph. Since each K_n is a subgraph, certainly each C_n is subgraph for all $n \ge 3$.

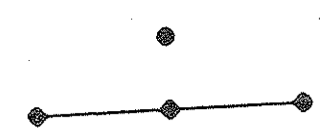
(c) This graph is locally finite; although the degrees of the vertices can be arbitrarily large, no vertex has an infinite number of neighbors. Clearly K_n is subgraph for all n, but note that K_∞ is not a subgraph. Again, since each K_n is a subgraph, certainly each C_n is subgraph.

31. (a) In order to state this definition, we will call the two centers of the stars a and b, and we will call the vertices of degree 1 emanating from these centers a_i and b_j, where $1 \le i \le m$ and $1 \le j \le n$. Then we can say that the double star $S(m,n)$ is a graph whose vertex set is $V = \{a, b, a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_n\}$ with edges ab, aa_i for all i, and bb_j for all j.

(b) It is enough to identify the two parts. Let $V_1 = \{a, b_1, b_2, \ldots, b_n\}$ and $V_2 = \{b, a_1, a_2, \ldots, a_m\}$. It is clear that no edge joins two vertices in the same part.

33. The parts are the set of all bit strings of length n with an odd number of 1's and the set of all bit strings of length n with an even number of 1's. Clearly if two bit strings differ in exactly one bit (which is how two vertices get to be adjacent in Q_n), then one of them must have one more 1 than the other, so that one of them is in V_1 and the other is in V_2.

35. (a) If v is an isolated vertex, then v is incident to (i.e., an element of, since edges are sets of vertices) no edge; in symbols, $v \notin e$ holds for all edges e. Thus the proposition that G *has* an isolated vertex is $\exists v \in V : \forall e \in E : v \notin e$. The following picture shows a graph with an isolated vertex.



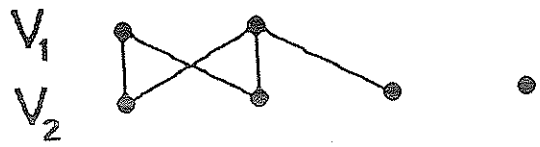
(b) To say that K_3 is a subgraph is to say that there exist three distinct vertices v_1, v_2, and v_3, each adjacent to the others. To say that K_4 is not a subgraph is to say the negation of the similar statement for four vertices. Thus the proposition is $(\exists v_1 \in V:$

$\exists v_2 \in V : \exists v_3 \in V : (\{v_1, v_2\} \in E \wedge \{v_1, v_3\} \in E \wedge \{v_2, v_3\} \in E)) \wedge \neg(\exists v_1 \in V : \exists v_2 \in V : \exists v_3 \in V : \exists v_4 \in V : (\{v_1, v_2\} \in E \wedge \{v_1, v_3\} \in E \wedge \{v_1, v_4\} \in E \wedge \{v_2, v_3\} \in E \wedge \{v_2, v_4\} \in E \wedge \{v_3, v_4\} \in E))$. The following graph satisfies this proposition.

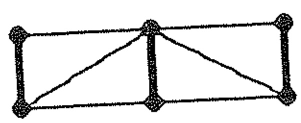
(c) We just need to state that every edge joins vertices in different parts:

$$\forall e \in E : \exists v_1 \in V_1 : \exists v_2 \in V_2 : e = \{v_1, v_2\}.$$

The following graph is bipartite with parts V_1 and V_2.



(d) A subgraph such as this is called a matching. A matching can be thought of as a subset of the set of edges, such that each vertex is incident to exactly one edge in the subset. Thus our proposition is $\exists E' \subseteq E : \forall v \in V : \exists! e \in E' : v \in e$. In the graph depicted here, the matching is shown with heavy lines.



37. We claim that any such graph G is a disjoint union of cycles. To see this, suppose that we start at any vertex v of G and "walk" along the edges. Since each vertex has degree 2, our walk will continue until we return to v. When the walk has been thus concluded, we have completed one cycle (a subgraph C_n for some n), and no other edges can be incident to vertices in that cycle. If any vertices remain, we repeat this walking process and find another cycle, disjoint from the first. We continue in this way until all the vertices have been traversed.

39. The largest complete subgraph K_r in G is called a maximal clique of G. In order to find its size, we recursively find the size of a maximal clique in the graph G with its last vertex deleted and the size of a maximal clique in the subgraph consisting of all neighbors of the last vertex. The larger of the former and one greater than the latter is clearly the size of a maximal clique in G, since any clique of G must either omit the last vertex or contain it (and in the latter case contain only other vertices adjacent to it). This algorithm is given below in pseudocode.

```
procedure maxclique(G : graph with vertices v1, v2, ..., vn)
    if n = 1 then return(1)
    else
        begin
            H1 ← G with vn deleted
            H2 ← G with vn and all vertices not adjacent to vn deleted
                {in each case, edges are kept if their endpoints remain}
            return(max(maxclique(H1), 1 + maxclique(H2)))
        end
```

SECTION 8.2 Traveling Through a Graph

1. (a) This sequence is none of these. It is not a walk, since fg is not an edge. Since it is not a walk, none of the other terms can possibly apply either.

(b) This is a walk of length 8; there is an edge joining each successive pair of vertices in the sequence. It is also a trail, since no edge is repeated. It is not closed, since it does not end at the same vertex at which it began. (In particular, it is not a cycle.) Finally, it is not a path, since the vertex a, for example, is visited twice.

(c) This is a walk of length 2, specifically a closed walk, since it ends at the same vertex (d) at which it began. It is not a trail (or any of the other things listed), since edge df is used twice.

(d) This is a walk of length 0. The terms trail and path also apply, but this walk is not closed, since our definition required a closed walk to have length at least 1.

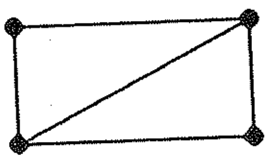
(e) This sequence is a cycle of length 6: It ends where it begins and otherwise uses no vertex more than once. Clearly the terms walk, closed walk, trail, and closed trail also apply.

(f) This walk of length 7 is closed. It repeats no edge, so it is a (closed) trail. It does repeat a vertex (d), so it is not a cycle.

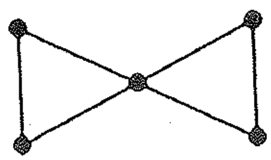
(g) This is a walk of length 7. None of the other terms apply, since edge bd is used twice.

3. Many answers are possible. We just need to use each edge exactly once. These Euler tours can be found by trial and error or by using Algorithm 1. Two Euler tours are a, h, i, j, d, i, c, h, j, f, e, d, c, f, g, a, g, c, b, a and a, h, j, i, h, c, i, d, j, f, e, d, c, f, g, a, g, c, b, a.

5. (a) We can form a graph containing a Hamilton cycle but no Euler tour by starting with a cycle itself and then adding one extra edge, thereby destroying the property that each vertex has even degree. The following picture shows such a graph.



(b) A graph such as the one shown below can have no Hamilton cycle, since a Hamilton cycle would have to visit the middle vertex four times. It does have an Euler tour, since the degree of each vertex is even.



(c) Clearly any cycle C_n has both a Hamilton cycle and an Euler tour (in fact it is its own such cycle and tour).

(d) The graph consisting of just the path shown here clearly has neither a Hamilton cycle nor an Euler tour.

7. None of these statements is necessarily true. The two graphs depicted below will provide counterexamples. In both cases G is the subgraph shown in heavy lines, and H is the entire graph.

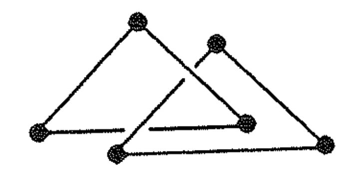
(a) In the graph shown on the left, G is connected, but H is not.

(b) In the graph shown on the right, H is connected, but G is not.

(c) In the graph shown on the right, G has two components, but H has only one.

(d) In the graph shown on the left, H has two components, but G has only one.

9. Consider the graph consisting of two components, each one a copy of K_3. We can model this graph with two triangles of string. It might happen that the triangles are linked (one triangle loops through the other). In that case, if we pick up one vertex of one of the triangles, then the entire graph is "attached." The heuristic does not work in such a case. This situation is illustrated in the following diagram.



11. Assume that G is not connected. Let V_1 be the set of vertices in one component of G, say having k vertices, where $1 \le k \le n-1$; and let V_2 be the remaining vertices of G. Then V_2 has $n-k$ vertices. The most edges that G can have is therefore $C(k,2)+C(n-k,2)$, which would happen if there were just two components, each one a complete graph. But this sum equals $[k(k-1)/2]+[(n-k)(n-k-1)/2]$, which simplifies to $\frac{1}{2}(2k^2-2nk+(n^2-n))$. This quadratic function of k attains its maximum at one of the endpoints of the interval $[1, n-1]$, since the graph of this function is a right side up parabola. If $k=1$, then this expression yields $\frac{1}{2}(2-2n+(n^2-n)) = \frac{1}{2}(n-1)(n-2)$. If $n=k-1$, the same value is obtained. Therefore the answer to the question is $\frac{1}{2}(n-1)(n-2)$. Note that it is not a correct proof to automatically *assume* that the maximum occurs when just one vertex of G is isolated. It turned out that the maximum we found is obtained with this arrangement, but we had to prove that this was so. At first glance it is conceivable that the maximum might occur for some other splitting of the vertices of G into components.

13. **(a)** There is no path from a vertex other than a to vertex a, so vertex a (together with its loop) is in a strong component by itself. Similarly, there is no path from vertex b to any other vertex, so vertex b is in a strong component by itself. Similarly each of e and f is in a component by itself. Vertices c and d can each be reached from the other by a path (in fact, by an edge), but there is no path from either of these vertices to any other vertices of the digraph. Therefore these two vertices and the two edges connecting them form a strong component. Note that edges ab and ec lie in no strong component.

(b) It is clear that each of the vertices 40, 50, 60, and 70 can be reached from the others by traveling around the cycle, but that there is no path from any of these vertices to 20 or 30, nor from 80 or 90 to any of these vertices. Therefore vertices 40, 50, 60, and 70 and the four edges connecting them form one strong component. Each of the other vertices is in a strong component by itself.

(c) Vertex a forms one strong component (since there is no path from b or c to a). Vertices b and c and the four edges involving only them form the other strong component.

15. Let $v = v_0, e_1, v_1, \ldots, e_n, v_n = v$ be a given closed trail starting and ending at v. If all the vertices $v_1, v_2, \ldots, v_n$ are distinct, then we are done. If not, we let v_i and v_j, with $0 < i < j$ be a pair of equal vertices in the trail. Excising the portion of the trail from v_i to v_j (deleting v_i but not v_j), we obtain a shorter trail. If this trail is a cycle, we are done. Otherwise we repeat the process. We continue in this manner until no equal vertices remain in our trail (except for the first equaling the last), at which point we have the desired cycle.

17. Let u and v be distinct vertices in the digraph. By definition there are walks from u to v and from v to u. Concatenating these walks gives a closed walk $u = u_0, u_1, \ldots, u_n = u$ from u to itself. Let j be the smallest natural number such that $u_j = u_i$ for some $i < j$; clearly $j \leq n$, and the i for which $u_i = u_j$ is uniquely determined. Then the vertices $u_{i+1}, u_{i+2}, \ldots, u_j$ are all distinct, so $u_i, u_{i+1}, \ldots, u_j$ is the desired cycle.

19. By the Corollary to Theorem 1 in Section 8.1, the hypothesis is false. Hence any conclusion follows. (This is a vacuous proof.)

21. In each case we use the fact that a connected graph with at least one edge contains an Euler tour if and only if the degree of each vertex is even.

(a) The degree of each vertex in K_n is $n-1$. This is even if and only if n is odd. Thus K_n contains an Euler tour if and only if n is odd (and greater than 1, to guarantee at least one edge in the graph).

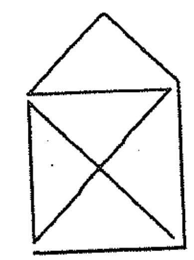
(b) The degrees of the vertices in $K_{m,n}$ are m and n. Thus $K_{m,n}$ contains an Euler tour if and only if both m and n are even positive integers.

(c) Clearly for all $n \geq 3$, the n-cycle is its own Euler tour.

(d) The vertices of Q_n are the bit strings of length n, and each vertex is adjacent to each of the n other vertices that differs from it in exactly one bit. Therefore the degrees of the vertices are even if and only if n is even. Therefore Q_n has an Euler tour if and only if n is even (and greater than 0, to guarantee at least one edge in the graph).

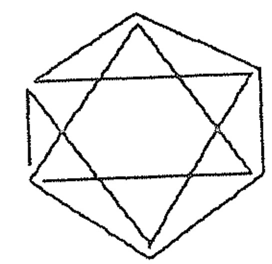
23. Such a tracing is exactly an Euler trail; each edge of the pseudograph must be traced out exactly once. We can apply the result obtained in Exercise 22 to determine whether such tracings are possible: A tracing exists if and only if there are zero or two vertices of odd degree.

(a) There are two vertices of odd degree in this graph, namely the vertices at the bottom. Therefore there is an Euler trail starting at one of these bottom vertices and ending at the other. The following figure shows one such trail.



(b) This graph has no tracing, since there are six vertices of odd degree, rather than none or only two.

(c) There are no vertices of odd degree in this graph, so there is an Euler tour. The following figure shows one such tour.



25. In order to begin to form the first closed trail C_1, we had to start with an edge. If there are no edges, then this cannot be done. In fact the conclusion is false if G has no edges. In this case necessarily $G = K_1$, and since by definition a closed trail must contain at least one edge, we are not able to conclude that G has an Euler tour.

27. We claim that there is a Hamilton cycle if and only if $m = n \geq 2$. The sufficiency of this condition is clear, since $u_1, v_1, u_2, v_2, \ldots, u_n, v_n, u_1$ (using the obvious notation) is a Hamilton cycle in $K_{n,n}$ for $n \geq 2$. Conversely, certainly $K_{1,1}$ has no Hamilton cycle, and if $m \neq n$ then $K_{m,n}$ can contain no Hamilton cycle, since any cycle must visit each part of $K_{m,n}$ equally often.

29. Our proof is very similar to the proof in the case considered in the text; we just have to turn the arrows around. Since G is strongly connected, there is a walk from u to v_1, and hence, by Theorem 1, a path from u to v_1. This path enters C for the first time at some vertex v_j, having begun $u = u_1, u_2, \ldots, u_l, v_j$, where the u_i's are distinct vertices not in C. But now we see a longer cycle in G, namely $v_1, v_2, \ldots, v_{j-1}, u_1, u_2, \ldots, u_l, v_j, v_{j+1}, \ldots, v_k, v_1$. This contradiction shows that C must have been a Hamilton cycle. (If $j = 1$, then this cycle is $v_1, v_2, \ldots, v_k, u_1, u_2, \ldots, u_l, v_1$.)

31. If G is bipartite, then every cycle must have even length, since it constantly jumps from one part to the other and eventually ends up where it began. We prove the converse by induction. Suppose that G has no cycles of odd length. If G has one or two vertices, then clearly G is bipartite. Assume the inductive hypothesis, and let G be a graph with $n+1$ vertices, containing no odd cycles. Let v be a vertex of G, and let $v_1, v_2, \ldots, v_r$ be the vertices adjacent to v. Let G' be the graph obtained by removing v from G. By the inductive hypothesis, G' (which still has no odd cycles) is bipartite. Choose the parts V_1 and V_2 so that for as large a k as possible, $v_1, v_2, \ldots, v_k$ are in V_1. If $k < r$, there must be a path from v_{k+1}, which is in V_2, to some $v_i \in V_1$, necessarily of odd length (otherwise we could put v_{k+1} into V_1 as well by switching parts for all vertices in the same component as v_{k+1}). But then the cycle consisting of this path, followed by edge v_iv, followed by edge vv_{k+1}, is an odd cycle, contradicting the hypothesis. Therefore $k = r$, and we can put v into V_2 to obtain a bipartition of G.

33. (a) It is easy to see that the sequence 0011 satisfies the condition.

(b) In order to construct a de Bruijn sequence for $n = 4$, we may as well start with four 0's. Following the pattern displayed in the $n = 3$ case, we might guess to try four 1's next. If we do so, then it is not hard with a little trial and error to come up with the following sequence: 0000111100101101. It is straightforward to check that all 16 four-bit patterns appear in this arrangement (viewed circularly, of course).

(c) Following the hint, we construct a graph with 2^{n-1} vertices and 2^n edges. By Exercise 32, there is an Euler tour in this digraph, since each vertex has in-degree and out-degree 2. The labels on this tour are the desired de Bruijn sequence. To see this, consider any n-bit string, and let v be the string consisting of its first $n-1$ bits. The n labels on edges in the portion of this tour ending at vertex v must always be exactly v, and v occurs twice in the tour—once followed by an edge labeled 0 and once followed by an edge labeled 1. Thus the given bit string occurs in the tour.

SECTION 8.3 Graph Representation and Graph Isomorphism

1. In each case we construct a matrix labeled by the vertices, and we put either a 1 or a 0 into the (i,j)th entry in the matrix if there is or is not an edge from vertex i to vertex j, respectively.

(a) The (a,c) entry is 1 (as is the (c,a) entry), for instance, since there is an edge between a and c; but since there is no edge joining a and e, the (a,e) entry and the (e,a) entry are both 0. Note here that the isolated vertex b has only 0's in its row and column.

$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

(b) Note how the structure of this matrix displays the isolated nature of vertex e.

$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

3. Our digraph must have vertices a, b, c, d, and e (the labels on the matrix), and edges between pairs of vertices for which the corresponding entry in the matrix is a 1. There is an edge from c to b, for instance, since the (c,b) entry is 1, but there is no edge from b to c, since the (b,c) entry is 0.