Module 3 Trees

Contents

3.1 Definitions and characterizations 72

3.2 Number of trees (Optional) 75

- Cayley's formula 77
- Kirchoff-matrix-tree theorem 79

3.3 Minimum spanning trees 83

- Kruskal's algorithm 85
- Prim's algorithm 87

Exercises 90

Trees are omnipresent in computer science and network analysis. The earliest mention of trees appear in the works of Kirchoff (1847), Jordan (1869), Sylvester (1882) and Cayley (1957).

3.1 Definitions and characterizations

Definitions.

- A connected acyclic graph is called a ***tree***.
- An acyclic graph is called a ***forest***.

  Thus, every component of a forest is a tree.

All the non-isomorphic trees up to seven vertices are shown below.

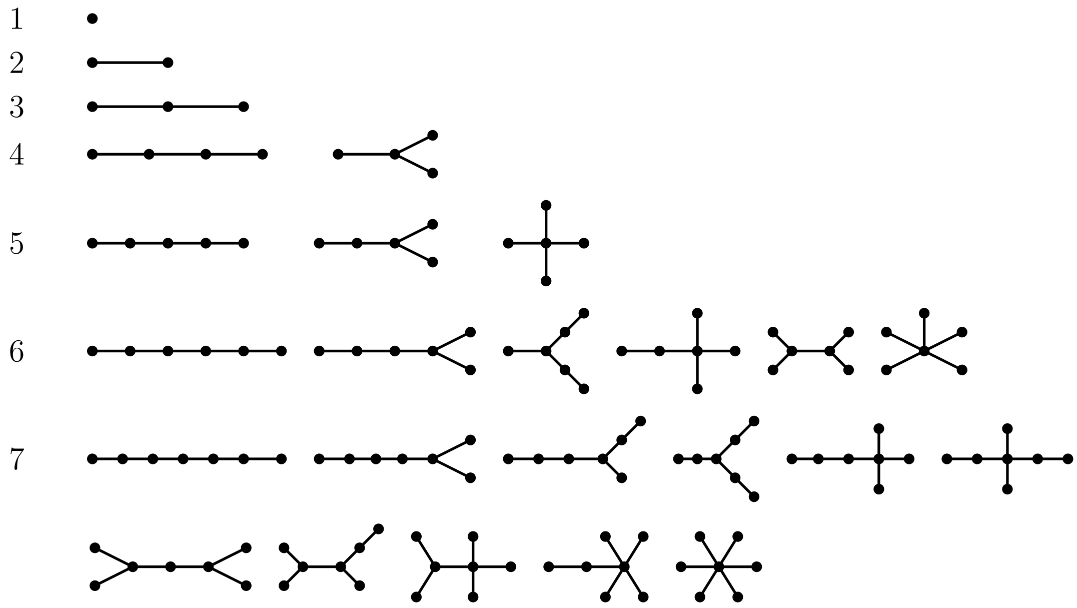


Figure 3.1: Non-isomorphic trees up to seven vertices.

Because of their simple structure, trees can be characterized in many ways. Some are proved below and some are included in the exercise list.

Theorem 3.1. *For any graph G, the following statements are equivalent.*

(a) G is a tree.

(b) G is connected and $m = n - 1$.

(c) G is acyclic and $m = n - 1$.

Proof. $(a) \Rightarrow (b)$: Since G is a tree, it is connected and therefore $m \geq n - 1$, by Corollary to Theorem 2.6. If $m \geq n$, then G contains a cycle, by Theorem 2.3, contrary to our hypothesis. Hence $m = n - 1$.

$(b) \Rightarrow (c)$: We have to only show that G is acyclic. On the contrary, assume that G contains a cycle. If e is an edge of this cycle, then $G - e$ is connected (by Theorem 2.15), and it has n vertices and $n - 2$ edges, a contradiction to Corollary of Theorem 2.6.

$(c) \Rightarrow (a)$: Since $m = n - 1$, G is connected or contains a cycle by Theorem 2.7. But by our hypothesis, G is acyclic. Therefore, G is connected and acyclic, and so it is a tree. □

Theorem 3.2. *A graph G is a tree if and only if any two vertices are connected by a unique path.*

Proof. (1) G is a tree $\Rightarrow$ Any two vertices are connected by a unique path.

Any two vertices are connected by a path, since G is connected. Next suppose that there exist two vertices u and v which are connected by two paths $P(u, v)$ and $Q(u, v)$. They may be internally disjoint or they may have a common vertex. In the latter case, let $z \neq v$ be the last common vertex. (It is possible that $z = u$.) Then $P(z, v)$ and $Q(z, v)$ are two internally disjoint (z, v)-paths which together form a cycle in G, a contradiction to the acyclicity of G.

(2) Any two vertices are connected by a unique path $\Rightarrow$ G is a tree.

Clearly, G is connected. We next claim that G is acyclic. Suppose G contains a cycle C. If we choose any two vertices u, v on this cycle, they are connected by two paths, a contradiction to our hypothesis. □

Theorem 3.3. *A connected graph G is a tree if and only if every edge is a cut-edge.*

Proof. It is a consequence of Theorems 3.2 and 2.15. □

Theorem 3.4. *Every connected graph G contains a subgraph which is a spanning tree.*

Proof. By successively deleting an edge of a cycle as long as there are cycles, we get a spanning tree of G. □

Theorem 3.5. *Every tree T on n vertices $(n \geq 2)$ contains at least two vertices of degree 1.*

Proof. Let p be the number of vertices of degree 1 in T. Then

$$2(n-1) = 2m = \sum_{v \in V(T)} deg(v) \geq 2(n-p) + p.$$

Hence, $p \geq 2$. □

Previous two theorems imply that every connected graph G on n vertices contains a tree on k vertices for every k, $1 \leq k \leq n$. However, G need not contain *every* tree on k vertices, for a given k, $1 \leq k \leq n$. See Figure 3.2.

The following result gives a sufficient condition for a graph G to contain every tree on a given number of vertices.

Theorem 3.6. *(Chvatal, 1977) Let T be any tree on $k+1$ vertices. If $\delta(G) \geq k$, then G contains a tree (isomorphic to) T.*

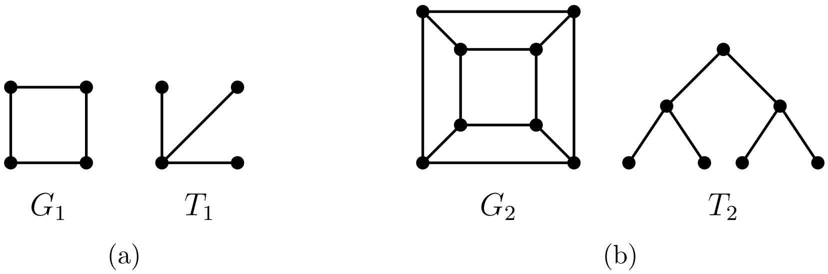


Figure 3.2: $T_1 \not\subseteq G_1$ and $T_2 \not\subseteq G_2$.

Proof. We prove the result by induction on k.

Basic step: If $k = 0$, the result is obvious.

Induction step ($k \geq 1$): Assume that if T' is any tree on k vertices and $\delta(G) \geq k-1$, then G contains T'. We show that if T is any tree on $k+1$ vertices and $\delta(G) \geq k$, then G contains T. By Theorem 3.5, T contains a vertex z of degree 1. Let y be the unique vertex adjacent with z. $T - z$ is a tree on k vertices. By induction hypothesis, G contains $T - z$. Since $deg_G(y) \geq \delta(G) \geq k+1$ and $T - z$ contains k vertices, we conclude that y is adjacent with a vertex x in $V(G) - V(T-z)$; see Figure 3.3. Then the tree $T - z + (x, y)$ is isomorphic with T. □

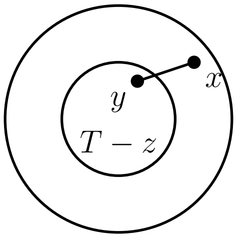


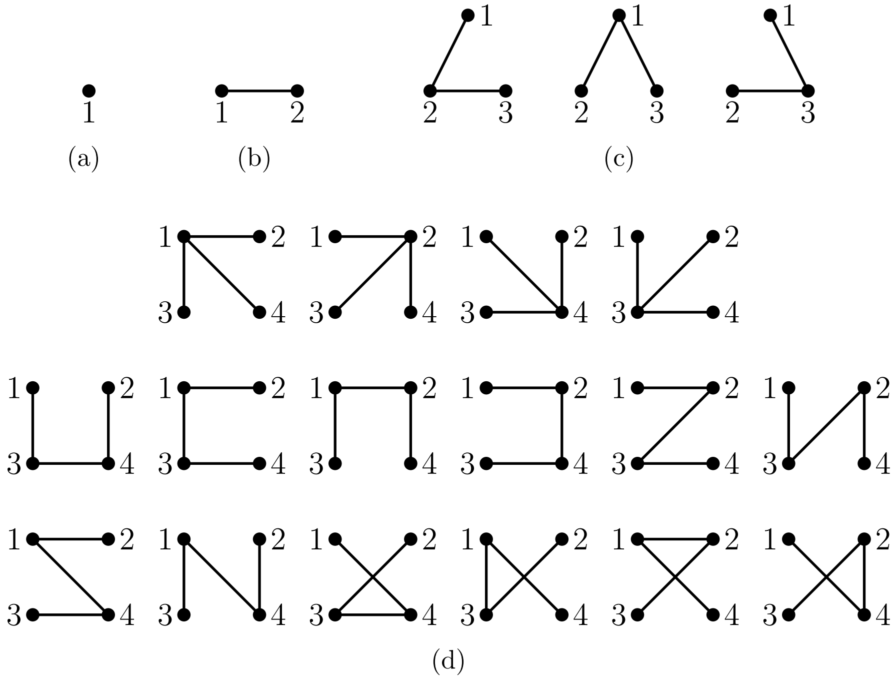
Figure 3.3: Construction of T from $T - z$.

3.2 Number of trees (Optional)

In this section, we are concerned with the following problem.

> How many trees are there on a given set of n vertices, where two trees T_1 and T_2 are counted distinct if $E(T_1) \neq E(T_2)$?

For simplicity, we assume that the vertices are $1, 2, \ldots, n$. And denote the number of trees on n vertices by $\mathcal{J}_n$. All the trees on $n = 1$ to 4 are shown below. Therefore $\mathcal{J}_1 = 1$, $\mathcal{J}_2 = 1$, $\mathcal{J}_3 = 3$, $\mathcal{J}_4 = 16$.



(a) (b) (c) (d)

Figure 3.4: Distinct trees upto four vertices.

It is important to note that $\mathcal{J}_n$ is not the number of non-isomorphic trees on n vertices. For example, $\mathcal{J}_3 = 3$, whereas there is only one non-isomorphic tree on three vertices; and $\mathcal{J}_4 = 16$, whereas there are only two non-isomorphic trees on four vertices. In Figure 3.4d, first four trees are mutually isomorphic and next twelve trees are mutually isomorphic.

• Cayley's formula

In 1889, Arthur Cayley proved a formula for the number of distinct trees on n given vertices. This he did towards his research on counting chemical isomers C_nH_{2n+2}. Since then there have been at least 20 new proofs. We give a proof due to Prüfer. This proof technique is widely used in discrete mathematics. Suppose we would like to count the number of objects in a given set S. Instead of counting the objects in S, we find a set T whose cardinality we know. We then establish a bijection from S to T which shows that $|S| = |T|$. Instead of establishing a bijection, we may establish an injection from S to T and an injection from T to S and appeal to Shroeder-Bernstein theorem and conclude that $|S| = |T|$.

Theorem 3.7 (Cayley's formula for the number of distinct trees)**.** *The number of distinct trees on n vertices is n^{n-2}.*

Proof. (Prüfer, 1918) It is an indirect proof. Let $S_n = \{(t_1, t_2, \ldots, t_{n-2}) : t_i \in \{1, 2, \ldots, n\}\}$. Clearly, $|S_n| = n^{n-2}$. Let F_n be the set of all distinct trees on n vertices; so $|F_n| = \mathcal{J}_n$. We first show that there is a one-one map from F_n to S_n, and next show that there is one-one map from S_n to F_n. It follows that there exists a one-one and onto map from F_n to S_n, and so $\mathcal{J}_n = |F_n| = |S_n| = n^{n-2}$.

(1) One-one map from F_n to S_n:

With a given tree $T \in F_n$, we associate a unique $(n-2)$-tuple $t = (t_1, t_2, \ldots, t_{n-2})$ by defining $t_1, t_2, \ldots, t_{n-2}$.

Definition of t_1: Among all the vertices of degree one in T, let s_1 be the vertex with the least label. We designate the unique vertex that is adjacent with s_1 as t_1.

Definition of t_2: Consider the tree $T - s_1$. Among all the vertices of degree one in $T - s_1$, let s_2 be the vertex with the least label. We designate the unique vertex that is adjacent with s_2 as t_2.

Definition of t_3: Consider the tree $T - s_1 - s_2$. Among all the vertices of degree one in $T - s_1 - s_2$, let s_3 be the vertex with the least label. We designate the unique vertex adjacent with s_3 as t_3.

This process is continued until t_{n-2} is defined.

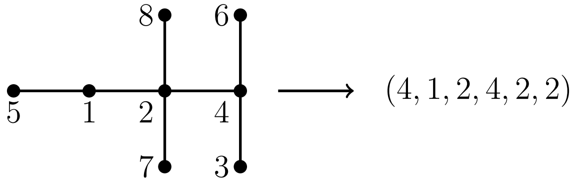


Figure 3.5: A tree T and its Prüfer code.

This association of t with T is a one-one map since t is unique, and moreover if $T = T'$, then $E(T) = E(T')$ and so $t = t'$.

- t is called the ***Prüfer code*** of T.

(2) One-one map from S_n to F_n:

Given a $(n-2)$-tuple $t = (t_1, t_2, \ldots, t_{n-2}) \in S_n$, we construct a unique tree $T \in F_n$. We start with a forest F of n isolated vertices $1, 2, \ldots, n$ and proceed to add $n - 1$ edges, one at a time, to obtain T. Let $N = \{1, 2, \ldots, n\}$.

Addition of first edge: Let $s_1 \in N$ be the least element which does not appear in $(t_1, t_2, \ldots, t_{n-2})$. Join s_1 and t_1.

Addition of second edge: Consider $N - \{s_1\}$ and $(t_2, t_3, \ldots, t_{n-2})$. Let $s_2 \in N - \{s_1\}$ be the least element which does not appear in $(t_2, \ldots, t_{n-2})$. Join s_2 and t_2.

Addition of third edge: Consider $N - \{s_1, s_2\}$ and $(t_3, t_4, \ldots, t_{n-2})$. Let $s_3 \in N - \{s_1, s_2\}$ be the least element which does not appear in $(t_3, t_4, \ldots, t_{n-2})$. Join s_3 and t_3.

This process is continued until $s_{n-2} \in N - \{s_1, s_2, \ldots, s_{n-3}\}$ is identified and joined to t_{n-2}. Until now we have added $n-2$ edges to F and we are left with two elements $s_{n-1}, s_n \in N - \{s_1, s_2, \ldots, s_{n-2}\}$. Finally, join s_{n-1} and s_n. In Figure 3.6, we have illustrated this construction by taking the 6-tuple (4, 1, 2, 4, 2, 2) and get back the tree of Figure 3.5. □

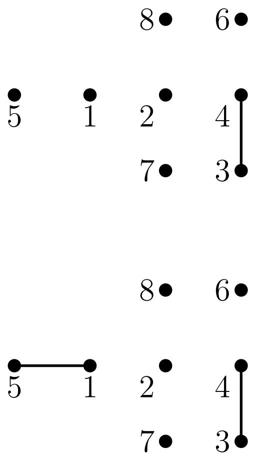
• Kirchoff-matrix-tree theorem

Kirchoff (1847) developed the theory of trees to find the current in each branch and around each circuit in an electrical network; any text book on network analysis contains these details. In this context, he was interested in finding the spanning trees of a given graph (representing an electrical network). Figure 3.7 shows 8 distinct spanning trees of a graph.

Remarks.

(1) The number of spanning trees of a graph G is denoted by $\mathcal{J}(G)$, where two spanning trees T_1 and T_2 are counted as distinct if $E(T_1) \neq E(T_2)$.

(2) $\mathcal{J}(G) = 0$ if and only if G is disconnected.

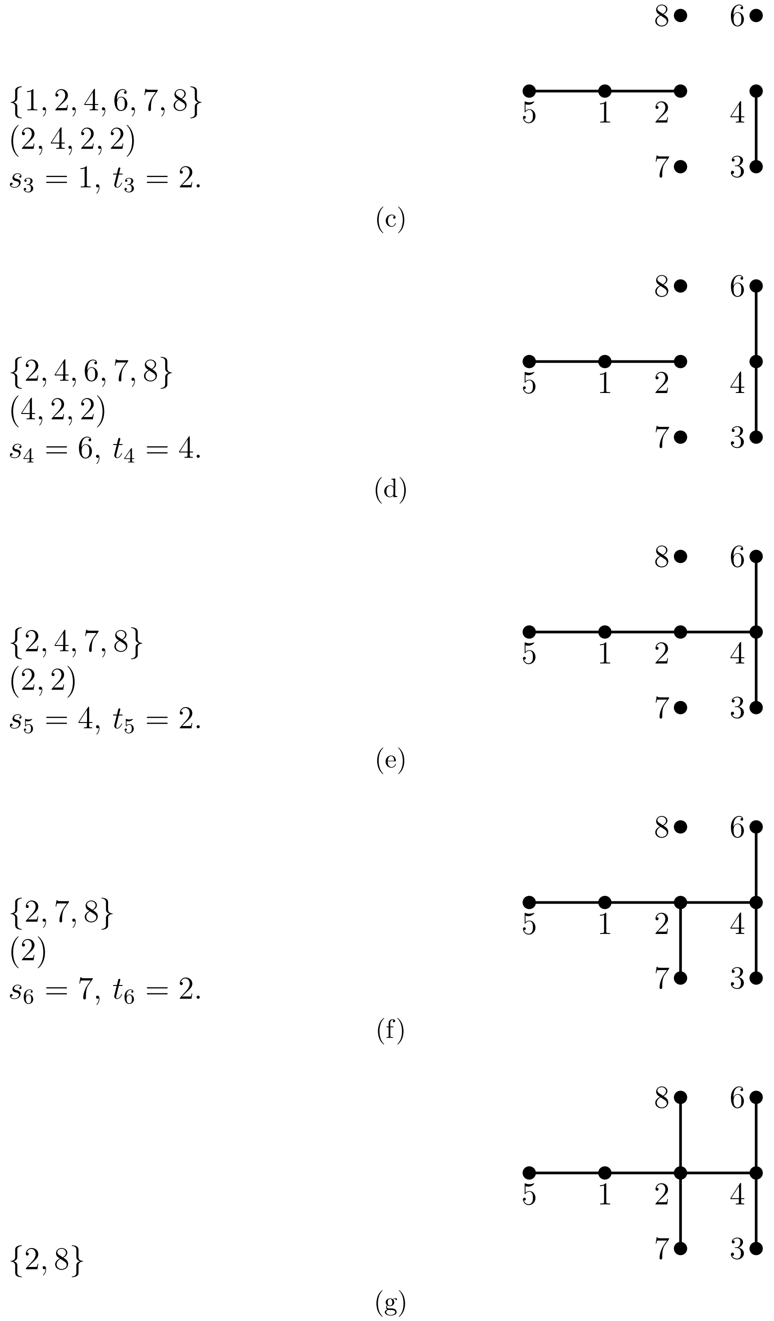


$\{1,2,3,4,5,6,7,8\}$
$(4,1,2,4,2,2)$
$s_1 = 3,\ t_1 = 4.$

(a)

$\{1,2,4,5,6,7,8\}$
$(1,2,4,2,2)$
$s_2 = 5,\ t_2 = 1.$

(b)



$\{1,2,4,6,7,8\}$
$(2,4,2,2)$
$s_3 = 1,\ t_3 = 2.$

(c)

$\{2,4,6,7,8\}$
$(4,2,2)$
$s_4 = 6,\ t_4 = 4.$

(d)

$\{2,4,7,8\}$
$(2,2)$
$s_5 = 4,\ t_5 = 2.$

(e)

$\{2,7,8\}$
(2)
$s_6 = 7,\ t_6 = 2.$

(f)

$\{2,8\}$

(g)

Figure 3.6: Construction of a tree with a given Prüfer code (4,1,2,4,2,2).

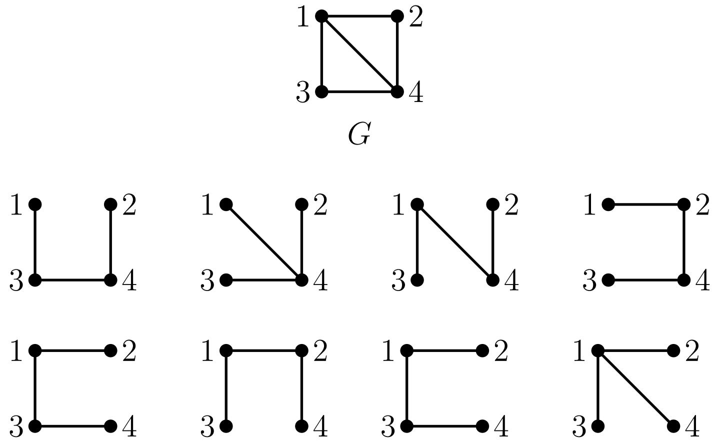


Figure 3.7: G and its eight distinct spanning trees

(3) $\mathcal{J}(G) = 1$ if and only if G is a tree.

(4) $\mathcal{J}(K_n) = \mathcal{J}_n = n^{n-2}$ (Cayley's formula).

(5) If $G^{(k)}$ denotes the graph obtained by replacing every edge of a simple graph G with k parallel edges, then $\mathcal{J}(G^{(k)}) = k^{n-1}\mathcal{J}(G)$.

(6) If G_0 denotes the simple graph obtained from a graph G by deleting all the loops of G, then $\mathcal{J}(G_0) = \mathcal{J}(G)$. So, in finding $\mathcal{J}(G)$ one can ignore the loops.

Kirchoff discovered an amazing formula for $\mathcal{J}(G)$. Cayley's formula can be derived using Kirchoff's formula.

Theorem 3.8 (Kirchoff-Matrix-Tree-Theorem). *Let G be a loopless graph and $A(G)$ be its adjacency matrix. Let $K(G)$ denote the $n \times n$ matrix obtained from $A(G)$ by applying the following two operations:*

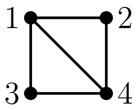
1. Change the sign of every off-diagonal entry from $+$ to $-$.

2. Replace the i-th diagonal entry by the degree of the i-th vertex.

Then all the $(n-1)\times(n-1)$-co-factors of $K(G)$ have the same value and the common value is $\mathcal{J}(G)$, that is $\mathcal{J}(G) = (-1)^{s+t}det(M_{(s,t)})$, where $M_{(s,t)}$ is the $(n-1)\times(n-1)$ matrix obtained from $K(G)$ by deleting the s-th row and the t-th column. □

- ○ $K(G)$ is called the ***Kirchoff matrix***.

Illustration 1:



(a) G

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

(b) $A(G)$

$$\begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}$$

(c) $K(G)$

$$\begin{bmatrix} 3 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 3 \end{bmatrix}$$

(d) $M(2,3)$

Figure 3.8: A graph G, its adjacency matrix, associated Kirchoff matrix and a co-factor.

$\mathcal{J}(G) = (-1)^{2+3}det(M(2,3)) = 8$. The eight distinct spanning trees of G are shown in Figure 3.7.

Illustration 2: Derivation of Cayley's formula using Kirchoff-Matrix-Tree Theorem.

Apply row/column operations to $M(1,1)$ and reduce it to the following triangular matrix (for example, $R_1 \leftarrow R_1+R_2+\cdots+R_{n-1}$, $R_i \leftarrow R_1+R_i$, for $2 \leq i \leq n-1$).

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & n & 0 & \dots & 0 \\ 0 & 0 & n & \dots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \dots & n \end{bmatrix}_{(n-1)\times(n-1)}$$

$\mathcal{J}(K_n) = (-1)^2 det(M(1,1)) = n^{n-2}$.

$$\begin{bmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 1 & 0 & \dots & 1 \\ & & \vdots & & \\ 1 & 1 & 1 & \dots & 0 \end{bmatrix}_{n\times n}$$

(a) $A(K_n)$

$$\begin{bmatrix} n-1 & -1 & -1 & \dots & -1 \\ -1 & n-1 & -1 & \dots & -1 \\ -1 & -1 & n-1 & \dots & -1 \\ & & \vdots & & \\ -1 & -1 & -1 & \dots & n-1 \end{bmatrix}_{n\times n}$$

(b) $K(K_n)$

$$\begin{bmatrix} n-1 & -1 & \dots & -1 \\ -1 & n-1 & \dots & -1 \\ & & \vdots & \\ -1 & -1 & \dots & n-1 \end{bmatrix}_{(n-1)\times(n-1)}$$

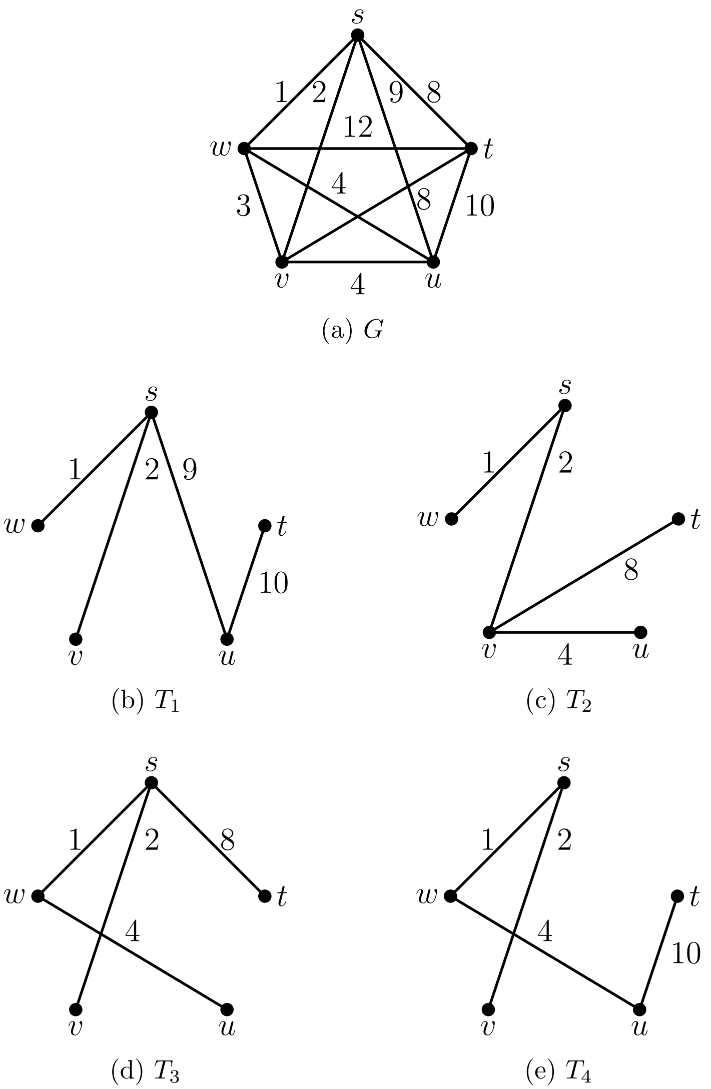
(c) $M(1,1)$

3.3 Minimum spanning trees

A real world problem whose algorithmic solution requires huge amount of time (technically, super exponential time) by brute force method but admits a decent (technically, low polynomial time) algorithm is the following. In fact, the algorithm follows "greedy paradigm". A few towns are to be connected by a network of roads. The cost of road construction between any two towns is given. The problem is to design a road network such that (i) given any two towns one is reachable from the other; and (ii) the total cost of the network is minimum possible.

It is clear that the problem can be easily modeled as a graph theoretic problem: Represent the towns by vertices and join two vertices by an edge e of weight $\mathcal{W}(e)$, if the cost of the road construction between the two corresponding towns is $\mathcal{W}(e)$. The problem is then to find a connected spanning graph H of a weighted graph G such that $\sum_{e\in H}\mathcal{W}(e)$ is minimum among all connected subgraphs H of G. We may assume that the weights are positive as they represent costs. It is immediate that such a H must be a tree; if H contains a cycle, then by deleting an edge of this cycle we get a connected subgraph H' of H whose weight is obviously less than that of H.

Definition. *A spanning tree of minimum weight in a connected weighted graph G is called a* ***minimum spanning tree*** *of G.*



(a) G

(b) T_1

(c) T_2

(d) T_3

(e) T_4

Figure 3.9: A weighted graph G and four of its spanning trees T_1, T_2,T_3, and T_4, with $\mathcal{W}(T_1) = 22$, $\mathcal{W}(T_2) = 15$, $\mathcal{W}(T_3) = 15$ and $\mathcal{W}(T_4) = 17$.

In the following we describe two algorithms to find a MST of a connected weighted graph G.

• Kruskal's algorithm (J.B. Kruskal, 1956)

Input: A weighted connected graph (G, $\mathcal{W}$)

Output: A minimum spanning tree T_k of G.

Step 1 (Initial): Arrange the edges of G in non-decreasing order, say π. Define H_0 to be the graph with vertex set $V(G)$ and no edges. (During the execution of the algorithm H_0 is updated by the addition of edges to yield T_k.)

Step 2: Select the first edge from π say e_1 and add it to H_0. Call the resulting graph H_1.

Step 3 (Recursion): After selecting i edges $e_1, e_2, \ldots, e_i$ and forming the graph H_i, find the next edge, say e_{i+1}, in the subsequence succeeding e_i in π which does not create a cycle with the edges already selected. Add it to H_i and call the resulting graph H_{i+1}.

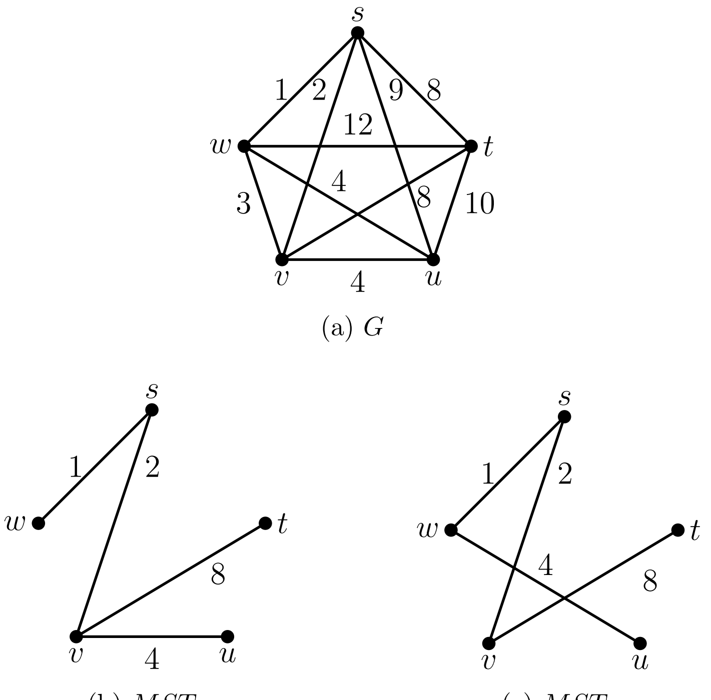
Step 4 (Stop Rule): Stop when $n-1$ edges are selected. Output H_{n-1} as T_k.

An illustration for Kruskal's algorithm

Consider the weighted graph shown in Figure 3.9. Arrange the edges in non-decreasing order with respect to their weight

$$((s,w),(s,v),(v,w),(v,u),(u,w),(s,t),(v,t),(s,u),(u,t),(t,w)).$$

As described in Kruskal's algorithm we successively select the edges (s,w), (s,v), (v,u) and (v,t), and obtain the first MST (MST_1) shown in Figure 3.10b. Note that at the third iteration, although the edge (v,w) is of minimum weight it cannot be selected as it creates a cycle with the two edges already selected.



(a) G

(b) MST_1

(c) MST_2

Figure 3.10: Two minimum spanning trees of G constructed by Kruskal's algorithm. Notice that both the trees have the same weight 15 but they are not isomorphic.

By choosing a different ordering of the edges $((s,w),(s,v),(v,w),(u,w),(v,u),$ $(v,t),(s,t),(s,u),(t,u),(t,w))$ and applying Kruskal's algorithm, we get the second minimum spanning tree shown in Figure 3.10c.

Theorem 3.9. *Any tree T_k constructed by Kruskal's algorithm is a MST of G.*

Proof. At the outset we observe that T_k is indeed a tree. This follows since H_{n-1} has $n-1$ edges and each H_i is acyclic (see step 3). Let T_{min} be a MST of G. If $E(T_k)$ $= E(T_{min})$, we are through. Next, suppose that $E(T_k) \neq E(T_{min})$ and let e_{i+1} be the first edge in T_k which is not in T_{min}. So, $E(T_k) = \{e_1, e_2, ..., e_i, e_{i+1}, ..., e_{n-1}\}$ and $E(T_{min}) = \{e_1, e_2, ..., e_i, f_{i+1}, ..., f_{n-1}\}$. Consider the graph $T_{min} + e_{i+1}$. It contains a unique cycle, say C; see Figure 3.11. Since $E(C) \nsubseteq E(T_k)$, there exists an edge $f \in E(C)$ which is in $E(T_{min})\ -E(T_k)$.

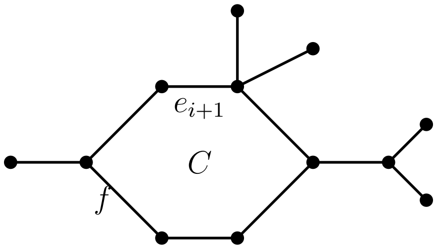


Figure 3.11: $T_{min} + e_{i+1}$.

Consider the tree $T_{min} + e_{i+1} - f$. We have $\mathcal{W}(T_{min}) \leq \mathcal{W}(T_{min} + e_{i+1} - f)$ $= \mathcal{W}(T_{min}) + \mathcal{W}(e_{i+1}) - \mathcal{W}(f)$. Therefore, $\mathcal{W}(f) \leq \mathcal{W}(e_{i+1})$. On the other hand, f too creates no cycles with the edges $e_1, e_2, \ldots, e_i$, since $e_1, e_2, \ldots, e_i, f \in T_{min}$. But we have selected e_{i+1} in the $(i+1)^{th}$ iteration. Therefore, $\mathcal{W}(e_{i+1}) \leq \mathcal{W}(f)$. Hence, $\mathcal{W}(e_{i+1}) = \mathcal{W}(f)$ and so $\mathcal{W}(T_{min}+e_{i+1}-f) = \mathcal{W}(T_{min})$. That is, $T_{min}+e_{i+1}-f$ is also a MST with one more edge in common with T_k. If $E(T_k) \neq E(T_{min} + e_{i+1} - f)$, we continue the above process enough number of times to conclude that T_k is a MST. □

• Prim's algorithm (R. C. Prim, 1957)

Input: A weighted connected graph (G, $\mathcal{W}$)

Output: A minimum spanning tree T_p of G.

Idea: If we have a subtree T of G and if we take a new vertex $x \in V(G) - V(T)$, and join it to precisely one vertex of T, then the resultant graph $T + x$ is also a tree.

Step 1: Select a vertex v_1 (arbitrarily). Select an edge of minimum weight in $[\{v_1\}, V - \{v_1\}]$, say (v_1, v_2). Define H_1 to be the tree with vertices v_1, v_2 and the edge (v_1, v_2).

Step 2: Having selected the vertices, $v_1, v_2, \ldots, v_k$ and $k-1$ edges, and forming the tree H_{k-1}, select an edge of minimum weight in $[\{v_1, \ldots, v_k\}, V - \{v_1, \ldots, v_k\}]$, say (v_i, v_{k+1}). Define H_k to be the tree obtained by joining v_{k+1} to $v_i \in H_{k-1}$.

Step 3 (Termination): Stop when $n-1$ edges are selected and output H_{n-1} as T_p.

An illustration:

Again consider the weighted K_5 shown in Figure 3.9. The successive selection of edges (v, s), (s, w), (u, w), and (t, v), yields the MST shown below.

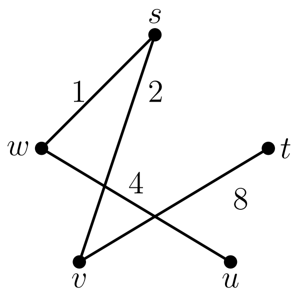


Figure 3.12: MST constructed using Prim's algorithm

Theorem 3.10. *Any tree T_p constructed by Prim's algorithm is a MST.*

Proof. As remarked in the description of the algorithm, T_p is indeed a tree. Let $e_1, e_2, \ldots, e_n$ be the order of edges in which they were selected to construct T_p. To prove the theorem it suffices to prove the following assertion:

(*) If there is a *MST* T with $e_1, e_2, \ldots, e_{k-1} \in T$ and $e_k \notin T$, then there is a *MST* T' with $e_1, e_2, \ldots, e_{k-1}, e_k \in T'$.

So, let T be a *MST* with $e_1, e_2, \ldots, e_{k-1} \in T$ and $e_k \notin T$. Since $e_k = (v_i, v_{k+1})$ and $v_i \in \{v_1, \ldots, v_k\}$, we have $v_{k+1} \in V - \{v_1, \ldots, v_k\}$; see Step 2. $T + e_k$ contains a unique cycle. So there exists an edge $f = (u, v)$, where $u \in \{v_1, v_2, \ldots, v_k\}$ and $v \in \{v_{k+1}, \ldots, v_p\}$; see Figure 3.13. Consider the tree $T' = T + e_k - f$. Since at the k-th stage we have preferred to select e_k, it follows that $\mathcal{W}(e_k) \leq \mathcal{W}(f)$. Hence

$\mathcal{W}(T+e_k-f) = \mathcal{W}(T) + \mathcal{W}(e_k) - \mathcal{W}(f) \leq \mathcal{W}(T)$. Since T is a MST, we have $\mathcal{W}(T+e_k-f) = \mathcal{W}(T)$. So $T+e_k-f$ is also a MST of G containing $e_1, e_2, \ldots, e_{k-1}, e_k$. Thus (*), and hence the theorem is proved.

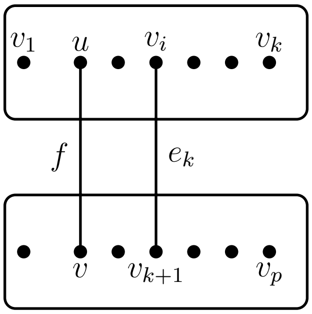


Figure 3.13: Construction of T' from $T + e_k$.

□

Exercises

1. Draw two non-isomorphic trees with the same degree sequence. What is the smallest n for the existence of such trees?

2. Suppose G is a simple connected graph such that $G - v$ is acyclic, for every vertex $v \in V(G)$. Determine the number of edges in G, and determine G itself.

3. If G is a tree with all the degrees odd, then show that the number of edges in G is odd.

4. Find a tree realization and a disconnected graph realization of the sequence $(1^6, 2, 3^2, 4)$.

5. (a) If a tree has degree sequence $(1^k, 4^p)$, then find k (in terms of p).

   (b) Which of the quizzes stated on the beginning of this course is now solved.

6. Let $n \geq 2$. Show that a sequence $(d_1, d_2, \ldots, d_n)$ of integers is the degree sequence of a tree iff (i) $d_i \geq 1$, for $1 \leq i \leq n$ and (ii) $\sum\limits_{i=1}^{n} d_i = 2n - 2$.

7. Let G be a forest with k components. How many edges should one add to G, so that the resulting graph is a tree?

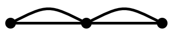
8. Show that there exists a connected graph with degree sequence $d_1 \geq d_2 \geq \ldots \geq d_n$ iff (i) $d_i \geq 1$, for $1 \leq i \leq n$ and (ii) $\sum\limits_{i=1}^{n} d_i \geq 2n - 2$.

9. Show that in a tree T there are at lest $\Delta(T)$ vertices of degree 1.

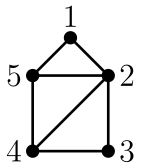
10. If G is a simple, 3-regular, connected graph, then show that in every spanning tree T of G, $n_1(T) = n_3(T) + 2$, where $n_i(T) =$ the number of vertices of degree i in T.

11. A tree T on 10 vertices has exactly one vertex of degree 4 and exactly one vertex of degree 3. Find the maximum degree of T and find all the possible degree sequence of T. Draw three such non-isomorphic trees.

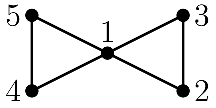
12. The maximum degree of a tree T is 5 and let $n_i(T)$ denote the number of vertices of degree i. If $n_1(T) = 50$ and $n_2(T) = n_3(T) = n_4(T) = n_5(T)$, find the number of vertices in T.

13. Let $P_n^{(2)}$ be the graph obtained by doubling every edge of P_n. Find the number of spanning trees of $P_n^{(2)}$. For example: $P_3^{(2)} :=$ 

14. Find $\mathcal{J}(G_1)$ and $\mathcal{J}(G_2)$, where G_1 and G_2 are shown below, using Kirchoff-matrix-tree theorem.



(a) G_1



(b) G_2

Trees are extensively used in data structures. These trees in fact have more structure. A vertex v of a tree T, chosen arbitrarily and distinguished from others, is called the ***root of*** T; T is called a ***rooted tree***. A rooted tree in which the root has degree 1 or 2 and every other vertex has degree ≤ 3 is called a ***binary tree***. These trees are extensively used for data representation. In general, an m***-ary tree*** is a rooted tree in which the root has degree $\leq m$ and every other vertex has degree $\leq m+1$. The vertices of degree 1, other than the root are called the ***leaves***. The largest distance between the root and a leaf is called the ***height of*** T.

A complete m-ary tree is a m-ary tree in which (i) the root has degree m, (ii) every other vertex has degree $m+1$ or 1 and (iii) all the leaves have the same distance from the root.

---

15. Let T be a rooted binary tree on n vertices in which exactly one vertex has degree 2 and every other vertex has degree 1 or 3. Show that: (a) n is odd, and (ii) the number of vertices of degree one is $\dfrac{n+1}{2}$.

16. Find the number of vertices in a complete m-ary tree.

17. Let T be an m-ary tree with root v of degree m and every other vertex of degree $m+1$ or 1. Show that the height of T with k leaves is at least $\log_m k$.

18. Show that the complete binary tree on 7 vertices is not a subgraph of the graph of the 3-cube Q_3.

19. A rooted binary tree T is said to be ***height-balanced***, if for every vertex $v \in V(T)$, the heights of the subtrees rooted at the left and right child of v differ by at most one; the difference is denoted by $b_T(v)$. So $b_T(v) = 0$ or 1. By convention, $b_T(v) = 0$, if v is a leaf.
    (a) Draw all the non-isomorphic height balanced trees of height $h = 0, 1, 2, 3$.
    (b) Find the number of vertices in a height balanced tree T with height h such that $b_T(v) = 0$, for every $v \in V(T)$.
    (c) An height-balanced tree T with $b_T(v) = 1$, for every non-leaf vertex $v \in V(T)$, is called the Fibonacci tree. If n_h denotes the number of vertices in a height balanced tree of height h. Show that $n_0 = 1$, $n_1 = 2$ and $n_h = 1 + n_{h-1} + n_{h-2}$, for $h \geq 2$.

20. Let G be a graph shown in Figure 3.14.

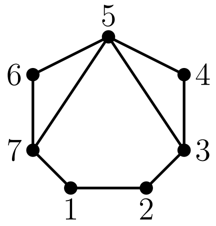


Figure 3.14: G

Draw the following spanning trees of G.
    (i) A spanning tree which is a complete binary tree.
    (ii) A spanning tree of diameter 5.
    (iii) A spanning tree T such that $dist_T(1, v) = dist_G(1, v)$, for every vertex v.

21. Which of the following spanning trees of $K_3 + K_4^c$ exist? Justify your answer.
    (i) A spanning tree which is a complete binary tree.
    (ii) A spanning tree of diameter 3.
    (iii) A spanning tree of diameter 5.

22. Let G be the graph shown in Figure 3.15. Draw a spanning tree T of G such that $dist_T(x, y) \leq 3 \cdot dist_G(x, y)$, for any two vertices x and y.

23. Prove or disprove: The two trees T_1 and T_2 with Prüfer codes $(1, 2, 1, 2, 1, 2)$ and $(1, 1, 1, 2, 2, 2)$ respectively are isomorphic.

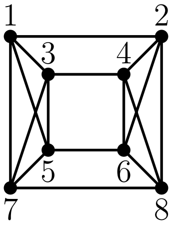


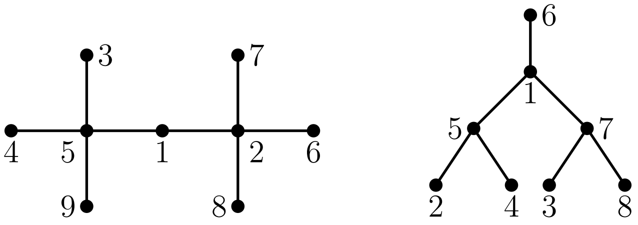
Figure 3.15: G

24. Draw the tree whose Prüfer code is $(6, 4, 1, 1, 5, 3)$.

25. What is the tree whose Prüfer code is $(k, k, \ldots k)$, for some integer k.

26. What is the tree whose Prüfer code is $(a, b, a, b, \ldots, a, b)$, for some integers a and b.

27. Find the Prüfer codes of the following trees.



(a) T_1 (b) T_2

28. (a) If a tree T has diameter 2, find the number of vertices of degree 1.
    (b) If a tree T has diameter 3, find the number of vertices of degree 1.
    (c) Let $n_1(T)$ denote the number of vertices of degree 1 in a tree T, and let $\mathcal{F}_4$ denote the family of all trees on n vertices with diameter 4. Find $\max\{n_1(T) : T \in \mathcal{F}_4\}$.

29. Find all k for which there exists a spanning tree of diameter k of the graph shown below (Figure 3.16):

30. Let $S \subseteq V(G)$ be such that $G - S$ is acyclic. Show that $|S| \geq \dfrac{m - n + 1}{\Delta - 1}$.

31. For which values of k, does there exist an acyclic graph with 17 vertices, 8 edges and k components? Draw two non-isomorphic graphs with these properties.

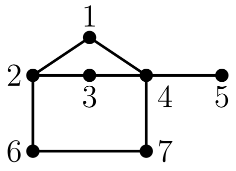


Figure 3.16: G

32. Let T be a tree on 21 vertices with degree sequence $(1^{15}, 3^k, 5^t, 6^1)$. Find k and t.

33. Show that there exists a tree with degree sequence $(1^t, 2, 3, \ldots, n)$ on $t + n - 1$ vertices iff $t - 3 = \dfrac{n(n-3)}{2}$.

34. Draw all the non-isomorphic trees which are isomorphic with their complements. Justify that you have drawn all the non-isomorphic trees.

35. Find the number of non-isomorphic trees with $n (\geq 6)$ vertices and maximum degree $n - 3$.

36. For which values of k, an acyclic graph with 17 vertices, 8 edges and k components exists? Draw two non-isomorphic graphs with these properties.

37. Show that every connected graph G contains a spanning tree T such that $diam(T) \leq 2 \cdot diam(G)$.

38. Show that a connected graph G is a tree iff $G + e$ contains a unique cycle, where e is any new edge.

39. Let G be a connected graph. Show that G contains a connected subgraph on k vertices, for every k, $1 \leq k \leq n$.

40. Let G be a connected graph and let $F \subseteq E(G)$. Show that there exists a spanning tree of G containing F iff $[F]$ is acyclic.

41. Let G be a connected graph and $S \subseteq V(G)$. Show that there exists a spanning tree T of G such that $deg_T(x) = deg_G(x)$, for every $x \in S$ iff $[S]$ and $[S, V - S]$ are acyclic.

42. Show that G is a forest iff every induced subgraph of G contains a vertex of degree ≤ 1.

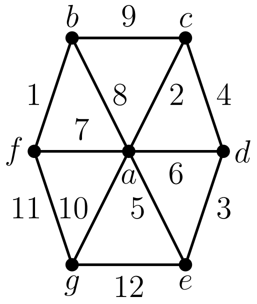
43. By using Kruskal' algorithm compute a minimum weight spanning tree of the graph whose weighted adjacency matrix is given below. Show the tree generated at every step of the algorithm.

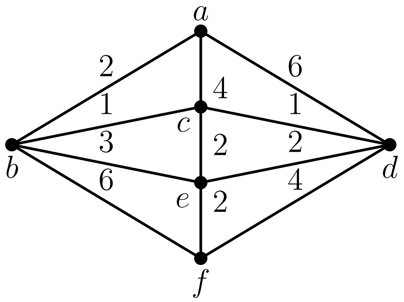
	a	b	c	d	e	f
a	0	10	6	0	0	0
b	10	0	5	0	3	2
c	6	5	0	4	0	1
d	0	0	4	0	4	0
e	0	3	0	4	0	7
f	0	2	1	0	7	0

44. Show that the following "dual" of Kruskal's algorithm outputs a minimum spanning tree.
Input: A weighted connected graph (G, $\mathcal{W}$).
Output: A minimum spanning tree T_k of G.
Step 1(Initial): Arrange the edges of G in non-increasing order, say π. Define H_0 to be the graph G. (During the execution of the algorithm the edges are deleted to yield T_k.)
Step 2: Select the first edge from π say e_1 such that $H_0 - e_1$ is connected and call the resulting graph H_1; if there is no such edge go to step 4.
Step 3 (Recursion): After deleting i edges say $e_1, e_2, \ldots, e_i$, and forming the graph H_i, find the next edge say e_{i+1} in the subsequence succeeding e_i in π such that $H_i - e_{i+1}$ is connected and call the resulting graph H_{i+1}; if there is no such edge go to step 4.
Step 4 (Termination): Stop when $m - n + 1$ edges are deleted from G. Output H_{m-n+1} as T_k.

45. Find a minimum spanning tree of the weighted graph shown below by using Prim's algorithm with g as the starting vertex. Draw all the intermediate trees which lead to MST. What is the total weight of MST?

46. Repeat the above problem for the graph shown below.





47. (a) Give an example of a tree T on 6 vertices such that T^c is disconnected.

    (b) Characterize the trees whose complements are disconnected.

48. Let $r, s(\geq 2)$ be integers. Given an example of a simple graph G on $(r-1)(s-1)$ vertices such that G contains no tree on r vertices and G^c contains no K_s.

49. A ***star*** is a tree on n vertices with one vertex of degree $n-1$ and $n-1$ vertices of degree 1. If T is a tree on $n(\geq 1)$ vertices which is not a star, then show that T^c contains an isomorphic copy of T.