Introduction to Graph Theory

George Voutsadakis[1]

[1]Mathematics and Computer Science
Lake Superior State University

LSSU Math 351

1 Trees
- Properties of Trees
- Counting Trees
- More Applications

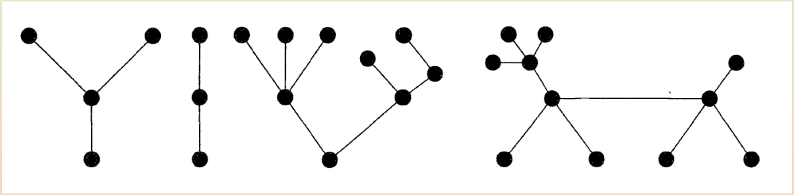
Subsection 1

Properties of Trees

Trees and Forests

- A **forest** is a graph that contains no cycles.
- A connected forest is a **tree**.

  Example: The figure shows a forest with four components, each of which is a tree.



- Note that trees and forests are simple graphs.

  Note: Because of their relatively simple structure, trees serve as testing ground for conjectures about general graphs.

  Several conjectures not proven yet for arbitrary graphs are known to be true for trees.

Characterizations of Trees

Theorem

Let T be a graph with n vertices. Then the following are equivalent:

(i) T is a tree;
(ii) T contains no cycles, and has $n-1$ edges;
(iii) T is connected, and has $n-1$ edges;
(iv) T is connected, and each edge is a bridge;
(v) Any two vertices of T are connected by exactly one path;
(vi) T contains no cycles, but the addition of any new edge creates exactly one cycle.

(i)$\Rightarrow$(ii): If T is a tree, then it contains no cycles and has $n-1$ edges.
If T is a tree it is by definition acyclic and connected.
We use induction on the number n of vertices to show that it must have $n-1$ edges.

- For $n=1$ the result is trivial.
- Assume the conclusion holds for any tree with k vertices, $1 \leq k < n$.

Characterizations of Trees (Cont'd)

(i)$\Rightarrow$(ii) (Cont'd):

- Let T be a tree with n vertices. Since T is acyclic and connected, the removal of any edge leaves two trees T_1 and T_2, with, say, k and $n-k$ vertices, respectively, where $1 \leq k < n$. By the Induction Hypothesis, T_1 has $k-1$ edges and T_2 has $n-k-1$ edges. Therefore, T has $(k-1)+(n-k-1)+1 = n-1$ edges.

(ii)$\Rightarrow$(iii): If T is acyclic with $n-1$ edges, then it is connected.

Suppose that T is disconnected and consists of k components $T_1, \ldots, T_k$, $k>1$, with, say, $n_1, \ldots, n_k$ vertices, respectively, $n_1+\cdots+n_k = n$. Then each component is acyclic and connected, i.e., a tree. By (i)$\Rightarrow$(ii), each component T_i has one fewer edge than it has vertices, i.e., component T_i has $n_i - 1$ edges. Therefore T has number of edges:

$$(n_1-1)+(n_2-1)+\cdots+(n_k-1) = (n_1+\cdots+n_k)-k = n-k \overset{k>1}{<} n-1.$$

But this contradicts the hypothesis that T has exactly $n-1$ edges.

Characterizations of Trees (Cont'd)

(iii)$\Rightarrow$(iv): If T is connected with $n-1$ edges, then each edge is a bridge.

Recall: A simple graph with n vertices and k components has at least $n-k$ edges.

Suppose that T has an edge which is not a bridge. Then the removal of that edge leaves a connected graph with n vertices. This graph must have at least $n-1$ edges. Thus, the original graph has at least $(n-1)+1=n$ edges, a contradiction.

Characterizations of Trees (Cont'd)

(iv)⇒(v): If T is connected and each edge is a bridge, then any two vertices of T are connected by exactly one path.

Since T is connected, any two vertices are connected by at least one path.

Suppose two vertices $x \neq y$ are connected by at least two paths, say

$$\begin{array}{ll} P_1: & x \to v_1 \to v_2 \to \cdots \to v_{k-1} \to y; \\ P_2: & x \to u_1 \to u_2 \to \cdots \to u_{\ell-1} \to y. \end{array}$$

Consider the first common vertex on P_1 and P_2 after x (such exists, since they share $y \neq x$), say $v_i = u_j$. Then the path

$$x \to v_1 \to v_2 \to \cdots \to v_i = u_j \to u_{j-1} \to \cdots \to u_1 \to x$$

is a cycle. But then $x \to v_1$ is not a bridge, a contradiction.

Characterizations of Trees (Conclusion)

(v)$\Rightarrow$(vi): If any two vertices of T are connected by exactly one path then:

(a) T contains no cycles;
(b) The addition of any new edge in T creates exactly one cycle.

(a) Suppose T contains a cycle, say $x \to v_1 \to v_2 \to \cdots \to v_{k-1} \to x$. Then the vertices x and v_1 are connected by two paths $P_1 : x \to v_1$ and $P_2 : x \to v_{k-1} \to \cdots \to v_2 \to v_1$, a contradiction.

(b) Suppose the addition of a new edge $x \to y$ creates no cycle. Then there is no path from x to y in T, contradicting the hypothesis. Suppose that the addition of a new edge $x \to y$ creates two cycles, say

$$\begin{array}{ll} C_1 : & x \to y \to v_1 \to \cdots \to v_{k-1} \to x; \\ C_2 : & x \to y \to u_1 \to \cdots \to u_{\ell-1} \to x. \end{array}$$

Then in T there are two different paths from y to x:
$P_1 : y \to v_1 \to \cdots v_{k-1} \to x$ and $P_2 : y \to u_1 \to \cdots \to u_{\ell-1} \to x$.
This contradicts the hypothesis.

Characterizations of Trees (Conclusion)

(vi)$\Rightarrow$(i): If T contains no cycles and the addition of any new edge creates exactly one cycle, then T is connected.

Suppose that T is disconnected.

Then, if we add to T any edge joining a vertex of one component to a vertex in another, no cycle is created.

This contradicts the hypothesis.

Corollary

If G is a forest with n vertices and k components, then G has $n - k$ edges.

- Apply Part (iii) of the theorem to each component of G.
- By the Handshaking Lemma, the sum of the degrees of the n vertices of a tree is equal to twice the number of edges $(= 2n - 2)$. Therefore:

  If $n > 2$, any tree on n vertices has at least two end-vertices.

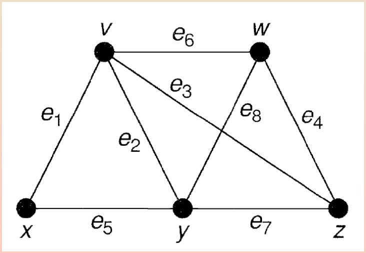
Spanning Trees

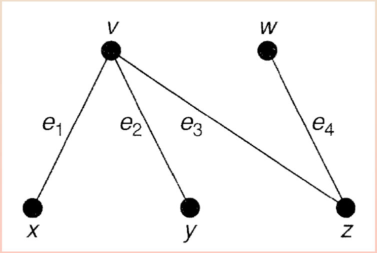
- Given any connected graph G, we can choose a cycle and remove any one of its edges, and the resulting graph remains connected.

  We repeat this procedure with one of the remaining cycles, continuing until there are no cycles left.

  The graph that remains is a tree that connects all the vertices of G. It is called a **spanning tree** of G.

  Example: Remove in turn e_5, e_6, e_7 and e_8.

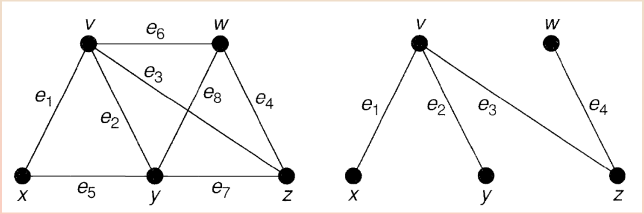




Spanning Forest, Cycle Rank and Cutset Rank

- If G is a graph with n vertices, m edges and k components, we can carry out the spanning tree procedure on each component of G.
- The result is called a **spanning forest**.
- The total number of edges removed in this process is the **cycle rank** of G, denoted by $\gamma(G)$. Note that $\gamma(G) = m - n + k$.
- We also define the **cutset rank** of G to be the number of edges in a spanning forest, denoted by $\xi(G)$. Note that $\xi(G) = n - k$.

  Example:



In the graph G above, $\gamma(G) = 4 = \xi(G)$.

Properties of Spanning Forests

- The **complement** of a spanning forest T of a (not necessarily simple) graph G is the graph obtained from G by removing the edges of T.

Theorem

If T is any spanning forest of a graph G, then:

(i) Each cutset of G has an edge in common with T;

(ii) Each cycle of G has an edge in common with the complement of T.

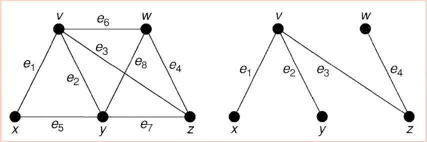
(i) Let C^* be a cutset of G. The removal of C^* splits a component of G into two subgraphs H and K. Since T is a spanning forest, T must contain an edge joining a vertex of H to a vertex of K. This edge is the required edge.

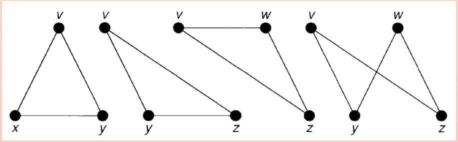
(ii) Let C be a cycle of G having no edge in common with the complement of T. Then C must be contained in T. This contradicts the acyclicity of T.

Fundamental Set of Cycles

- If we add to a spanning forest T of a graph G any edge of G not contained in T, then we obtain a unique cycle.
- The set of all cycles formed in this way, by adding separately each edge of G not contained in T, is the **fundamental set of cycles associated with T**.
- Sometimes we are not interested in the particular spanning forest chosen, and refer simply to a **fundamental set of cycles of G**.
- Note that the number of cycles in any fundamental set must equal the cycle rank $\gamma(G)$ of G.

  Example: For the graph G and spanning tree T shown on the left,



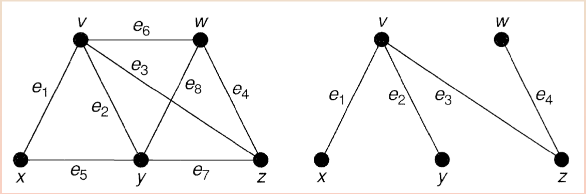


the fundamental set of cycles is shown on the right.

Fundamental Set of Cutsets

- The removal of any edge of a spanning forest T of a graph G divides the vertex set of T into two disjoint sets V_1 and V_2.
- The set of all edges of G joining a vertex of V_1 to one of V_2 is a cutset of G.
- The set of all cutsets obtained by removing separately each edge of T is the **fundamental set of cutsets associated with** T.
- Note that the number of cutsets in any fundamental set must equal the cutset rank $\xi(G)$ of G.

Example:



The fundamental set of cutsets of G associated with T consists of: $\{e_1, e_5\}$, $\{e_2, e_5, e_7, e_8\}$, $\{e_3, e_6, e_7, e_8\}$ and $\{e_4, e_6, e_8\}$.

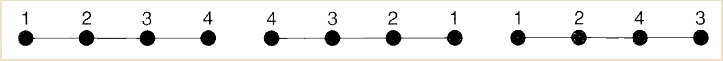
Subsection 2

Counting Trees

Example: Number of Labeled Trees

- We look at the number of labeled trees with a given number of vertices.

  Example: The figure shows three ways of labeling a tree with four vertices.

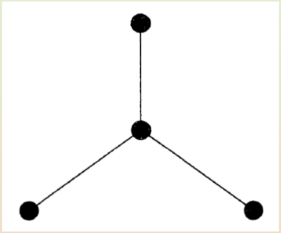
  

  - Since the second labeled tree is the reverse of the first one, these two labeled trees are the same.
  - Neither of the first two labeled trees is isomorphic to the third labeled tree, as can be seen by comparing the degrees of vertex 3.

  Since the reverse of any labeling does not result in a new one, the number of ways of labeling this tree is $\frac{4!}{2} = 12$.

Example (Cont'd)

- The number of ways of labeling the following tree is 4.



  - The central vertex can be labeled in four different ways;
  - Each one determines the labeling.
- The previous two shapes cover all possible unlabeled trees on four vertices.
- It follows that the total number of non-isomorphic labeled trees on four vertices is $12 + 4 = 16$.

Cayley's Theorem

Theorem (Cayley, 1889)

There are n^{n-2} distinct labeled trees on n vertices.

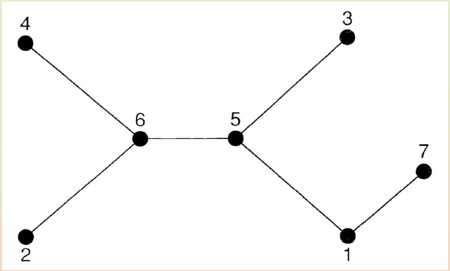
- We establish a one-one correspondence between the set of labeled trees of order n and the set of sequences $(a_1, a_2, \ldots, a_{n-2})$, where each a_i is an integer satisfying $1 \leq a_i \leq n$. Since there are precisely n^{n-2} such sequences, the result follows.

  We assume that $n \geq 3$, since the result is trivial if $n = 1$ or 2.

  Let T be a labeled tree of order n:
  - If b_1 is the smallest label assigned to an end-vertex, we let a_1 be the label of the vertex adjacent to the vertex b_1.
  - We then remove b_1 and its incident edge, leaving a labeled tree of order $n-1$. We let b_2 be the smallest label assigned to an end-vertex of the new tree, and a_2 be the label of the vertex adjacent to b_2.
  - We then remove the vertex b_2 and its incident edge, as before.
  - We proceed in this way until there are only two vertices left, and the required sequence is $(a_1, a_2, \ldots, a_{n-2})$.

Example of the Construction of the Sequence

Example: Let T be the labeled tree shown below.



- $b_1 = 2,\ a_1 = 6;$
- $b_2 = 3,\ a_2 = 5;$
- $b_3 = 4,\ a_3 = 6;$
- $b_4 = 6,\ a_4 = 5;$
- $b_5 = 5,\ a_5 = 1.$

The required sequence is therefore $(6, 5, 6, 5, 1)$.

Construction of the Tree

- To obtain the reverse correspondence, we take $(a_1, \ldots, a_{n-2})$:
  - let b_1 be the smallest number that does not appear in it, and join the vertices a_1 and b_1;
  - Then remove a_1 from the sequence, remove the number b_1 from consideration, and proceed as before.

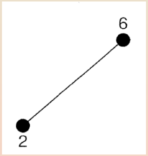
  In this way we build up the tree, edge by edge.
- We can check that, if we start with any labeled tree,
  - find the corresponding sequence,
  - and then find the labeled tree corresponding to that sequence,

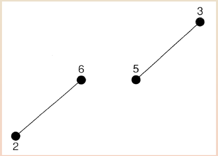
  we obtain the tree we started from.

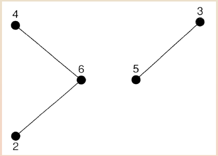
  We have now established the required correspondence, and Cayley's Theorem follows.

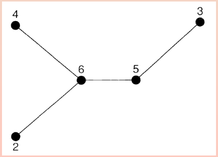
Example of the Construction of the Labeled Tree

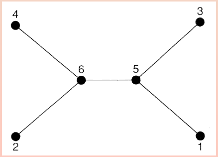
- Suppose we start with the sequence $(6,5,6,5,1)$.
- Then we build the labeled tree as follows:
  - $b_1 = 2$; Add edge 62; List: $(3,4,7)$; Sequence: $(5,6,5,1)$;
  - $b_2 = 3$; Add edge 53; List: $(4,7)$; Sequence: $(6,5,1)$;
  - $b_3 = 4$; Add edge 64; List: $(6,7)$; Sequence: $(5,1)$;
  - $b_4 = 6$; Add edge 56; List: $(5,7)$; Sequence: (1);
  - $b_5 = 5$; Add edge 15; List: $(1,7)$; Sequence: $()$;
  - Add edge 17.

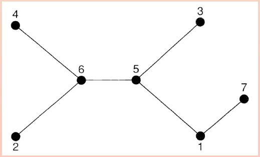










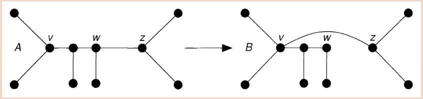


Second Proof of Cayley's Theorem

- Let $T(n,k)$ be the number of labeled trees on n vertices in which a given vertex v has degree k.

  We derive an expression for $T(n,k)$, and then sum from $k=1$ to $k=n-1$ to obtain the result.

  Let A be any labeled tree in which $\deg(v)=k-1$. The removal from A of any edge wz, not incident with v, leaves two subtrees, one containing v and either w or z (w, say), and the other z. If we join the vertices v and z, we obtain a labeled tree B in which $\deg(v)=k$.



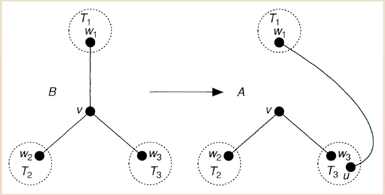
  We call a pair (A,B) of labeled trees a **linkage** if B can be obtained from A as described.

  A may be chosen in $T(n,k-1)$ ways; B is uniquely defined by the edge wz which may be chosen in $(n-1)-(k-1)=n-k$ ways; The total number of linkages (A,B) is $(n-k)T(n,k-1)$.

Counting Linkages

- Now let B be a labeled tree in which $\deg(v) = k$, and let $T_1, \ldots, T_k$ be the subtrees obtained from B by removing the vertex v and each edge incident with v.

  We obtain a labeled tree A with $\deg(v) = k - 1$ by removing from B just one of these edges (vw_i, say, where w_i lies in T_i), and joining w_i to any vertex u of any other subtree T.



  Note that the corresponding pair (A, B) of labeled trees is a linkage, and that all linkages may be obtained in this way.

  B can be chosen in $T(n, k)$ ways; The number of ways of joining w_i to vertices in any other T_j is $(n - 1) - n_i$, where n_i is the number of vertices of T_i; Thus, the total number of linkages (A, B) is
  $T(n, k)[(n - 1 - n_1) + \cdots + (n - 1 - n_k)] = (n - 1)(k - 1)T(n, k)$.

Counting Labeled Trees

- By the two countings of the total number of linkages, we get

$$(n-k)T(n,k-1) = (n-1)(k-1)T(n,k).$$

Noting that $T(n,n-1)=1$, we get:

$$\begin{aligned}
T(n,k) &= \tfrac{(n-1)k}{n-k-1}T(n,k+1) \\
&= \tfrac{(n-1)^2k(k-1)}{(n-k-1)(n-k-2)}T(n,k+2) \\
&= \tfrac{(n-1)^3k(k+1)(k+2)}{(n-k-1)(n-k-2)(n-k-3)}T(n,k+3) \\
&= \cdots \\
&= \tfrac{(n-1)^{n-k-1}k(k+1)(k+2)\cdots(k+(n-k-2))}{(n-k-1)(n-k-2)\cdots(n-k-(n-k-1))} \\
&\qquad T(n,k+(n-k+1)) \\
&= \tfrac{(n-1)^{n-k-1}(n-2)!}{(n-k-1)!(k-1)!} = \binom{n-2}{k-1}(n-1)^{n-k-1}.
\end{aligned}$$

Counting Labeled Trees (Cont'd)

- We found that

$$T(n,k) = \binom{n-2}{k-1}(n-1)^{n-k-1}.$$

So, summing over all possible values of k, we deduce that the number $T(n)$ of labeled trees on n vertices is

$$\begin{aligned} T(n) &= \sum_{k=1}^{n-1} T(n,k) \\ &= \sum_{k=1}^{n-1} \tbinom{n-2}{k-1}(n-1)^{n-k-1} \\ &= \{(n-1)+1\}^{n-2} \\ &= n^{n-2}. \end{aligned}$$

On the Number of Spanning Trees of K_n

Corollary

The number of spanning trees of K_n is n^{n-2}.

- To each labeled tree on n vertices there corresponds a unique spanning tree of K_n.

  Conversely, each spanning tree of K_n gives rise to a unique labeled tree on n vertices.

Subsection 3

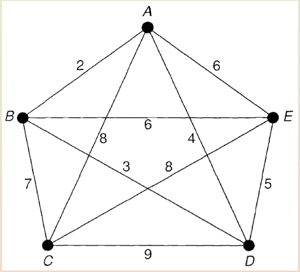
More Applications

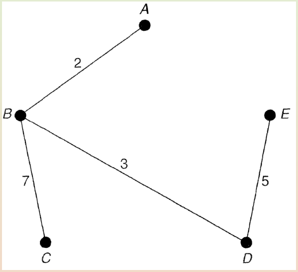
The Minimum Connector Problem

- Suppose that we wish to build a railway network connecting n given cities so that a passenger can travel from any city to any other.
- If, for economic reasons, the total amount of track must be a minimum, then the graph formed by taking the n cities as vertices and the connecting rails as edges must be a tree.
- The problem is to find an efficient algorithm for deciding which of the n^{n-2} possible trees connecting these cities uses the least amount of track, assuming that all pairwise distances are known.
- We can reformulate the problem in terms of weighted graphs:
  Denote the weight of the edge e by $w(e)$.
  The aim is to find the spanning tree T with least possible total weight $W(T)$.
- There is a simple **greedy algorithm** that solves the problem:
  It involves choosing edges of minimum weight in such a way that no cycle is created.

Illustrating the Algorithm

- Suppose there are five cities, as in the figure:





Choose edge AB (weight 2). Choose edge BD (weight 3). We cannot choose the edge AD (weight 4), since it would create the cycle ABD. Choose edge DE (weight 5). We cannot choose edges AE or BE (weight 6), since each would create a cycle. Choose edge BC (weight 7). This completes the tree.

Correctness of the Greedy Algorithm

Theorem

Let G be a connected graph with n vertices. Then the following construction gives a solution of the minimum connector problem:

(i) Let e_1 be an edge of G of smallest weight;

(ii) Define $e_2, e_3, \ldots, e_{n-1}$ by choosing at each stage a new edge of smallest possible weight that forms no cycle with previously chosen edges e_i.

The required spanning tree is the subgraph T of G whose edges are $e_1, \ldots, e_{n-1}$.

- The fact that T is a spanning tree of G follows immediately from the tree characterization theorem.
  It remains to show that the total weight of T is a minimum.
  Suppose, to the contrary, that S is a spanning tree of G such that $W(S) < W(T)$.

Correctness of the Greedy Algorithm (Cont'd)

- If e_k is the first edge in the above sequence that does not lie in S, then the subgraph of T formed by adding e_k to S contains a unique cycle C containing the edge e_k. Since C contains an edge e lying in S but not in T, the subgraph obtained from S on replacing e by e_k is a spanning tree S'.

  By the construction, $w(e_k) \leq w(e)$. So $W(S') \leq W(S)$. Moreover, S' has one more edge in common with T than S.

  By repeating this procedure, we can change S into T, one step at a time, with the total weight decreasing at each stage.

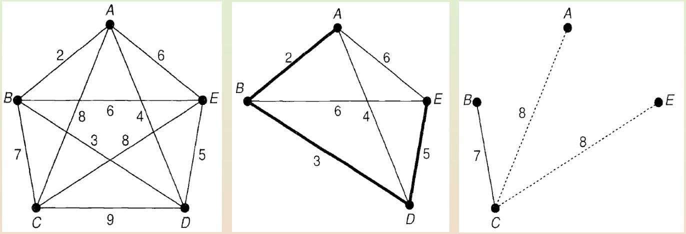
  This yields $W(T) \leq W(S)$, a contradiction.

Applying Greedy to Lower Bound Traveling Salesman

- We apply the greedy algorithm to obtain a lower bound for the solution of the traveling salesman problem.
- This is useful, since the greedy algorithm is an efficient algorithm, whereas no efficient general algorithms are known for the traveling salesman problem.
- If we take any Hamiltonian cycle in a weighted complete graph and remove any vertex v, then we obtain a semi-Hamiltonian path, and such a path must be a spanning tree.
- So any solution of the traveling salesman problem must consist of a spanning tree of this type together with two edges incident to v.
- It follows that if we take the weight of a minimum-weight spanning tree (obtained by the greedy algorithm) and add the two smallest weights of edges incident with v, then we get a lower bound for the solution of the traveling salesman problem.

Illustrating the Technique

- If we take the weighted graph of the figure on the left

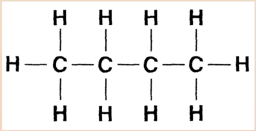


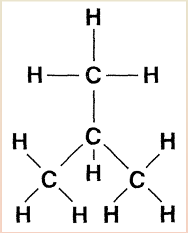
and remove the vertex C, then the remaining weighted graph has the four vertices A, B, D and E.
- The minimum weight spanning tree joining these four vertices is the tree whose edges are AB, BD and DE, with total weight 10.
- The two edges of minimum weight incident with C are CB and CA (or CE) with total weight 15.
- So, the lower bound for the traveling salesman problem is 25.

Enumeration of Chemical Molecules

- If we have a molecule consisting only of carbon atoms and hydrogen atoms, then we can represent it as a graph in which each carbon atom appears as a vertex of degree 4, and each hydrogen atom appears as a vertex of degree 1.

  Example: The graphs of *n*-butane and 2-methyl propane are shown below:





  Although they have the same chemical formula $\mathbf{C}_4\mathbf{H}_{10}$, they are different molecules because the atoms are arranged differently within the molecule.

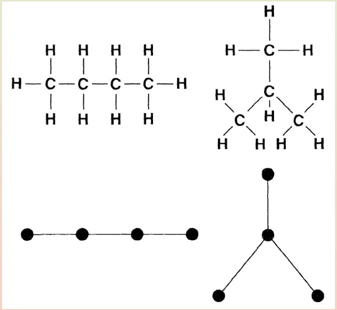
  These two molecules form part of a general class of molecules known as the **alkanes**, or **paraffins**, with chemical formula $\mathbf{C}_n\mathbf{H}_{2n+2}$.

  We want to find how many different molecules there are with this formula.

Counting the Alkanes

- The graph of a molecule with formula $\mathbf{C}_n\mathbf{H}_{2n+2}$ is connected and has:
  - $n+(2n+2)=3n+2$ vertices;
  - $\frac{4n+(2n+2)}{2}=3n+1$ edges.

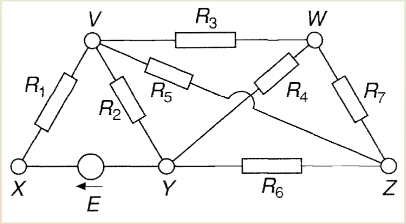
  Thus, it is a tree.
- Note also that the molecule is determined completely once we know how the carbon atoms are arranged.
  Discarding the hydrogen atoms, the problem reduces to that of finding the number of trees with n vertices, each of degree 4 or less.

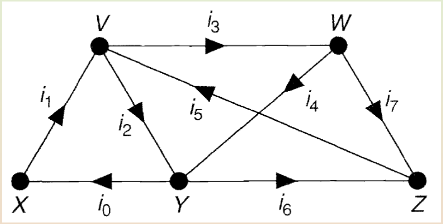


- This problem was solved by Cayley in 1875, by counting the number of ways in which trees can be built up from their center(s), but the argument is complicated.

Electrical Networks

- Suppose that we are given the electrical network

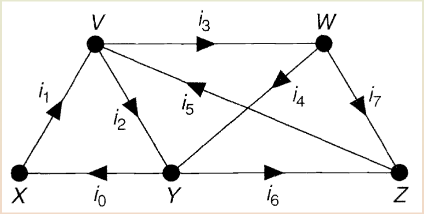




  and that we wish to find the current in each wire.
- To do this, we assign an arbitrary direction to the current in each wire and apply Kirchhoff's laws:
  - (i) the algebraic sum of the currents at each vertex is 0;
  - (ii) the total voltage in each cycle is obtained by adding the products of the currents i_k and resistances R_k in that cycle.

Application of Kirchhoff's Laws

- Applying Kirchhoff s second law to the cycles $VYXV$, $VWYV$ and $VWYXV$, we obtain the equations



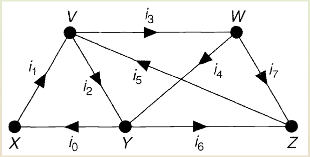
$$\begin{aligned} i_1R_1 + i_2R_2 &= E; \\ i_3R_3 + i_4R_4 - i_2R_2 &= 0; \\ i_1R_1 + i_3R_3 + i_4R_4 &= E. \end{aligned}$$

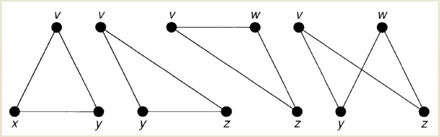
The last of these three equations is simply the sum of the first two, and gives us no further information.

Similarly, if we have the Kirchhoff equations for the cycles $VWYV$ and $WZYW$, then we can deduce the equation for the cycle $VWZYV$.

Eliminating Redundancy

- To find a set of cycles that gives us the information we need without any redundancy, we use a fundamental set of cycles





$$\begin{aligned}
&\text{for the cycle VYXV,} & i_1R_1 + i_2R_2 &= E\\
&\text{for the cycle VYZV,} & i_2R_2 + i_5R_5 + i_6R_6 &= 0\\
&\text{for the cycle VWZV,} & i_3R_3 + i_5R_5 + i_7R_7 &= 0\\
&\text{for the cycle VYWZV,} & i_2R_2 - i_4R_4 + i_5R_5 + i_7R_7 &= 0
\end{aligned}$$

The equations arising from Kirchhoff's first law are:

$$\begin{aligned}
&\text{for the vertex X,} & i_0 - i_1 &= 0\\
&\text{for the vertex V,} & i_1 - i_2 - i_3 + i_5 &= 0\\
&\text{for the vertex W,} & i_3 - i_4 - i_7 &= 0\\
&\text{for the vertex Z,} & i_5 - i_6 - i_7 &= 0.
\end{aligned}$$

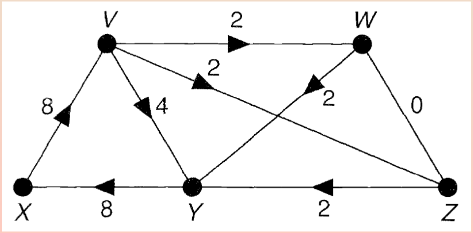
Finding the Currents

- These eight equations

$$\begin{aligned} i_1R_1 + i_2R_2 &= E & i_0 - i_1 &= 0 \\ i_2R_2 + i_5R_5 + i_6R_6 &= 0 & i_1 - i_2 - i_3 + i_5 &= 0 \\ i_3R_3 + i_5R_5 + i_7R_7 &= 0 & i_3 - i_4 - i_7 &= 0 \\ i_2R_2 - i_4R_4 + i_5R_5 + i_7R_7 &= 0 & i_5 - i_6 - i_7 &= 0 \end{aligned}$$

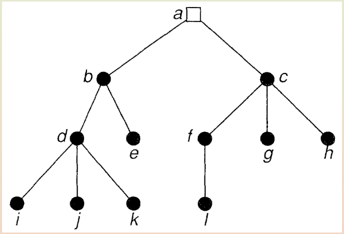
can now be solved to give the eight currents $i_0, i_1, \ldots, i_7$.

Example: If $E = 12$, and if each wire has unit resistance (that is, $R_i = 1$ for each i), then the solution is:



Searching Trees

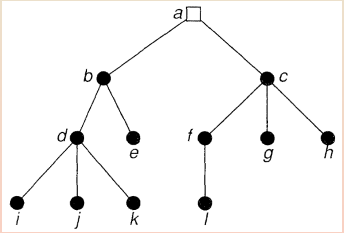
- In many applications, the trees that we consider have a hierarchical structure, with one vertex at the top (called the **root**), and the other vertices branching down from it:



- If a particular piece of information is required, we need to be able to search the tree in a systematic way. This involves examining every part of the tree until the desired vertex is found.
- We would like to find a search technique that eventually visits all parts of the tree without visiting any vertex too often.

Breadth First Search

- Two well-known search procedures - **depth first search** and **breadth first search** visit all the vertices, but in a different order:
    - In **breadth first search**, we fan out to as many vertices as possible, before penetrating deeper into the tree.
      This means that we visit all the vertices adjacent to the current vertex before proceeding to another vertex.
      Example:



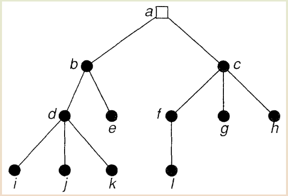
Start at vertex a. Visit the vertices b and c that are adjacent to a. Then visit the vertices d and e adjacent to b. Then the vertices f, g and h adjacent to c. Finally, visit the vertices i, j and k adjacent to d, and l adjacent to f.

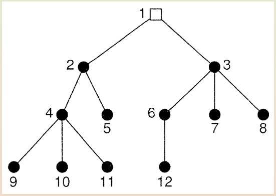
This gives us a labeled tree, where the labels correspond to the order in which the vertices are visited.

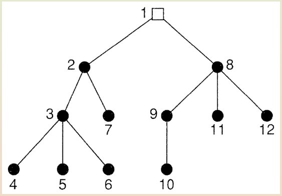
Depth First Search

- In **depth first search**, we penetrate as deeply as possible into a tree before fanning out to other vertices.

  Example: Consider again the tree on the left.

  Start at vertex a. Move down to b, d and i. Since we cannot penetrate further, we backtrack to d. Then go down to j. We must then backtrack again, and go to k. We now backtrack via d to b, from which we can go down to e. Backtracking to a then takes us to c, f and l. Then we visit g and h. Finally, we return to a.
- The labeled trees obtained from breadth- and from depth-first search, respectively, are shown on the right.