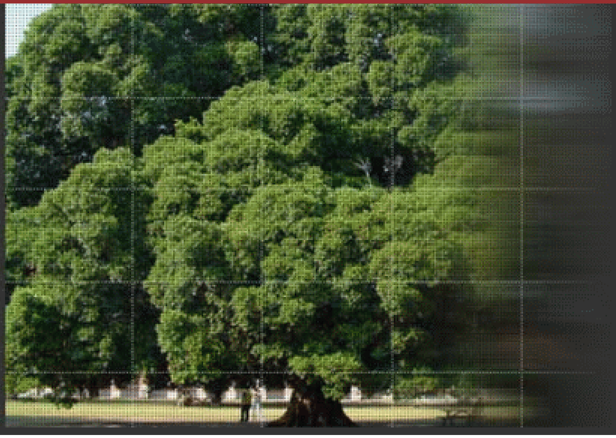
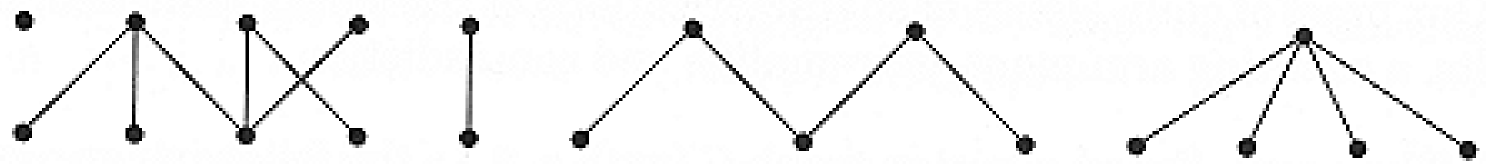


Chap 2 Trees and Distance

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Def:

- A graph with no cycle is *acyclic*.
- A *forest* is an acyclic graph.
- A *tree* is a connected acyclic graph.
- A *leaf* (or *pendant vertex*) is a vertex of degree 1.
- A *spanning subgraph* of G is a subgraph with vertex set $V(G)$.
- A *spanning tree* is a spanning subgarph that is a tree.



成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

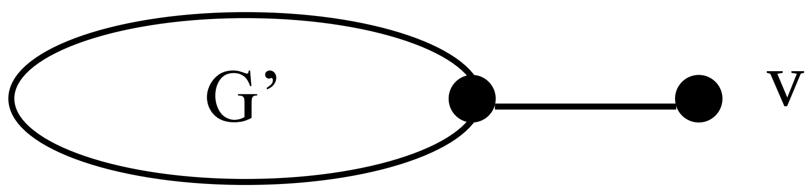
Lemma. Every tree with at least two vertices has at least two leaves. Deleting a leaf from an n-vertex tree produces a tree with n-1 vertices.

Proof:

- A connected graph with at least two vertices has an edge.

  In an acyclic graph, the endpoints of a maximal nontrivial path are leaves.
- Every u, w-path in G is also in G' ⇨ G' is connected.

  Since deleting a vertex cannot create a cycle ⇨ G' is acyclic.



COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Theorem. For an n-vertex graph G (with $n\geq 1$), the following are equivalent.

A) G is connected and no cycles.

B) G is connected and has n-1 edges.

C) G has n-1 edges and no cycles.

D) G has no loops and has, for exactly one u,v-path

Proof:

A$\Rightarrow${B,C}. By induction,

n=1. (trivial)

n>1. Given an acyclic connected graph G, $\exists$ a leaf v s.t.

$G'=G$-v also is acyclic and connected.

By I.H., $e(G') = n-2 \Rightarrow e(G) = n-1.$

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

A) G is connected and no cycles.

B) G is connected and has n-1 edges.

C) G has n-1 edges and no cycles.

Proof: B$\Rightarrow${A,C}.

Delete edges from cycles (no cut edges) of G one by one until the resulting graph G' is acyclic $\Rightarrow$ G' is connected $\Rightarrow$ $e(G')=n-1=e(G)$ $\Rightarrow$ G is acyclic.

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

A) G is connected and has no cycles.

B) G is connected and has n-1 edges.

C) G has n-1 edges and no cycles.

Proof. C⇒{A,B}

Let $G_1, G_2, \ldots, G_k$ be the components of G.

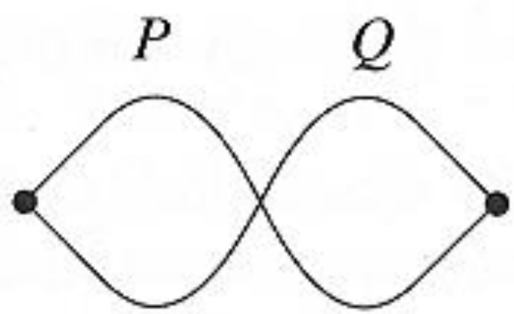
$e(G_i) = n(G_i) - 1 \Rightarrow e(G) = \sum_i [n(G_i) - 1] = n - k.$

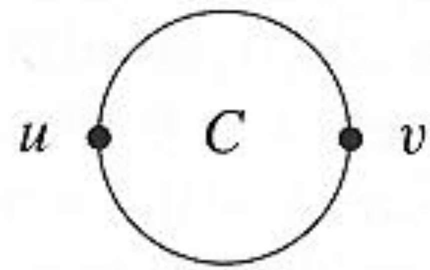
Since $e(G) = n - 1$, k=1 and G is connected.

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

A⇒D.

Since G is connected, each pair of vertices is connected by a path.





D ⇒A.

Clearly, G is connected.

If G has a cycle C,

then G has two u,v-paths for $u,v \in V(G) \Rightarrow G$ is acyclic.

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Corollary.

A) Every edge of a tree is a cut-edge.

B) Adding one edge to a tree forms exactly one cycle.

C) Every connected graph contains a spanning tree.

Proof.

A) By the fact that a tree has no cycles.

B) By the fact that a tree has a unique path linking each pair of vertices.

C) Iteratively deleting edges from cycles in a connected graph yields a connected acyclic subgraph.

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Proposition.

If T and T' are spanning trees of a connected graph G and $e \in E(T) - E(T')$, then there is an edge $e' \in E(T') - E(T)$ such that $T - e + e'$ is a spanning tree of G.

Proof.

Let U and U' be the two components of T-e.

Since T' is connected, T' has an edge e' with endpoints in U and U'.

T-e+e' is connected, has $n(G)$-1 edges, and is a spanning tree of G.

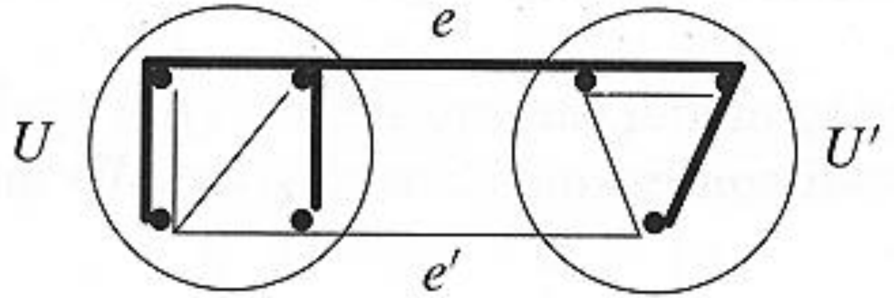
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Proposition.

If T and T' are spanning trees of a connected graph G and $e \in E(T) - E(T')$, then there is an edge $e' \in E(T') - E(T)$ such that $T' + e - e'$ is a spanning tree of G.

Proof.

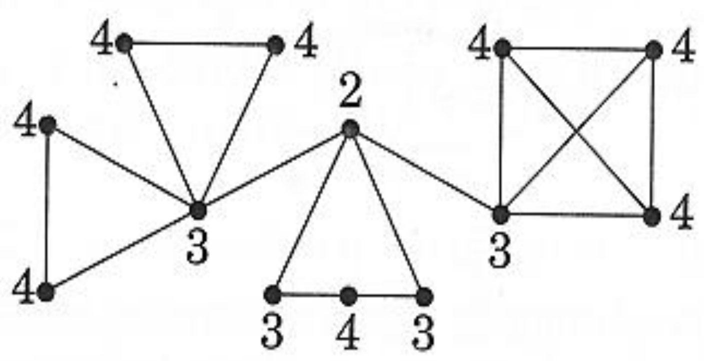
1. $T'+e$ contains a unique cycle C.
2. T is acyclic,
   Deleting e' breaks the only cycle in $T'+e$. Then,
   $T'+e-e'$ is connected and acyclic and is a spanning tree of G.



成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Def:

- *distance from u to v*, $d_G(u,v)$, is the least length of a u,v-path.
- *diameter* (diam G) is $\max_{u,v \in V(G)} d(u,v)$.
- *eccentricity of a vertex u*, $\varepsilon(u)$, is $\max_{v \in V(G)} d(u,v)$.
- radius of a graph G, rad G, is $\min_{u \in V(G)} \varepsilon(u)$.
- In the graph below, each vertex is labeled with its eccentricity. The radius is 2, the diameter is 4, and the length of the longest path is 7.



COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

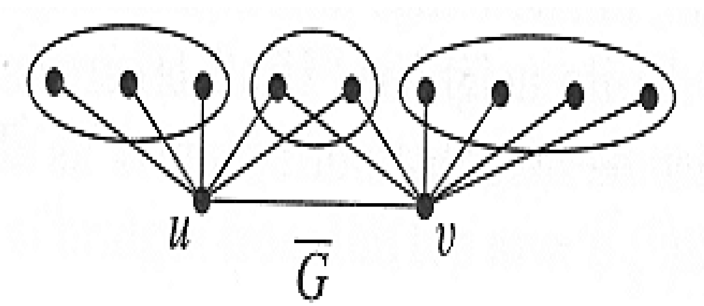
Theorem. If G is a simple graph, then $diamG \geq 3 \Rightarrow diam\overline{G} \leq 3.$

Proof:

When $diamG > 2$,there exist nonadjacent vertices $u, v \in V(G)$ with no common neighbor.

Every $x \in V(G) - \{u, v\}$ has at least one of $\{u,v\}$ as a nonneighbor$\Rightarrow$

x adjacent in $\overline{G}$ to at least one of $\{u, v\}$ in $\overline{G}$



成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- The *center* of a graph G is the subgraph induced by the vertices of minimum eccentricity.
- *Wiener index* of G is $D(G) = \sum_{u,v \in V(G)} d_G(u,v)$.

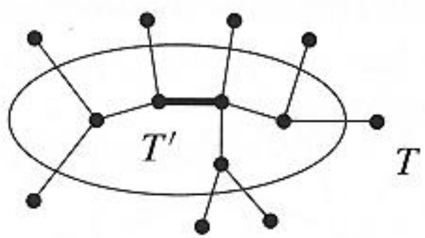
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Theorem. The center of a tree is a vertex or an edge.

Proof: By induction on the number of vertices in a tree *T*.

- Basis: $n(T) \le 2$. The center is the entire tree.
- Induction step: $n(T)>2$.

Form a tree *T'* by deleting every leaf of *T*.
The internal vertices on paths between leaves of *T* remain $\Rightarrow n(T') \ge 1$



$\varepsilon_{T'}(u) = \varepsilon_T(u) - 1, \forall u \in V(T')$

Also, the eccentricity of a leaf in *T* > the eccentricity of its neighbor in *T*.

$\therefore T$ and *T'* have the same center. By I.H., the result holds.

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Theorem. Among trees with n vertices, the Wiener index $D(T) = \sum_{u,v} d(u,v)$ is minimized by stars and maximized by paths, both uniquely.

Proof:

- A tree has n-1 edges, it has n-1 pairs of vertices at distance 1, and all other pairs have distance at least 2$\Rightarrow$ star minimizes $D(T)$.
- Uniqueness: Consider a leaf x in T with its neighbor v. If all other vertices have distance 2 from x, $x \in N(v) \Rightarrow T$ is a star.

$$D(K_{1,n-1}) = (n-1) + 2\binom{n-1}{2} = (n-1)^2$$

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

u: an endpoint of P_n

$\sum_{v \in V(P_n)} d(u,v) = \sum_{i=0}^{n-1} i = C(n,2)$

$D(P_n) = D(P_{n-1}) + C(n,2) \Rightarrow D(P_n) = C(n+1,3) \left(\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1} \right)$

Uniqueness: By induction on n

- Basis: n=1. The only tree with one vertex is P_1
- Induction step: n>1. Let u be a leaf.

$D(T) = D(T-u) + \sum_{v \in V(T)} d(u,v).$

By I.H., $D(T-u) \le D(P_{n-1})$, with equality iff T-u is a path

Consider the list of distance from u in P_n: 1,2,…,n-1

Any repetition makes $\sum_{v \in V(T)} d(u,v)$ smaller than when u is a leaf of a path.

T is a path and u is an endpoint of T.

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

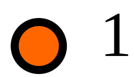
Lemma. If H is a subgraph of G, then $d_G(u, v) \le d_H(u, v)$

Corollary. If G is a connected n-vertex graph, then
$D(G) \le D(P_n)$

Proof:
Let T be a spanning tree of G. Then,
$D(G) \le D(T) \le D(P_n)$

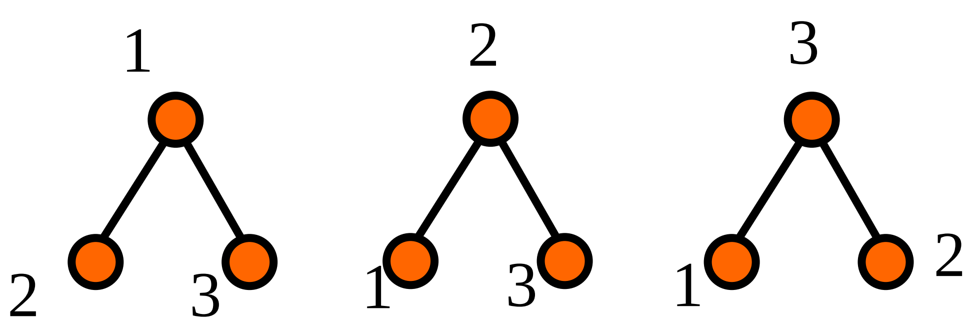
2.2. Spanning Trees and Enumeration

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY



one vertex

two vertices



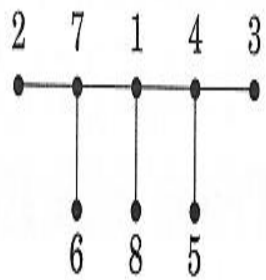
three vertices

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

(Prüfer code) Production of $f(T)=(a_1,\ldots,a_{n-2})$

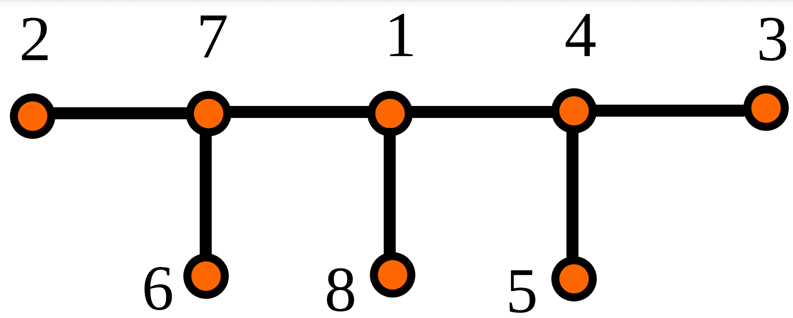
Input: A tree T with vertex set $S \subseteq N$

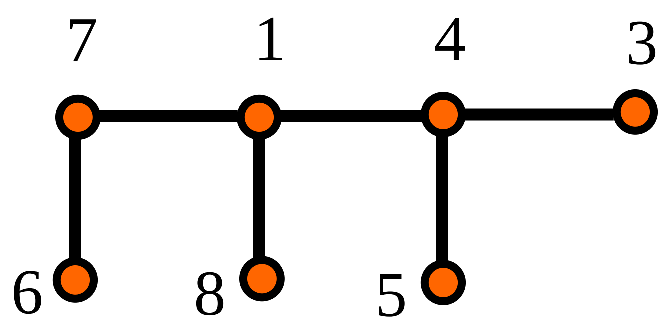
Iteration: At the ith step, delete the least remaining leaf, and let a_i be the neighbor of this leaf



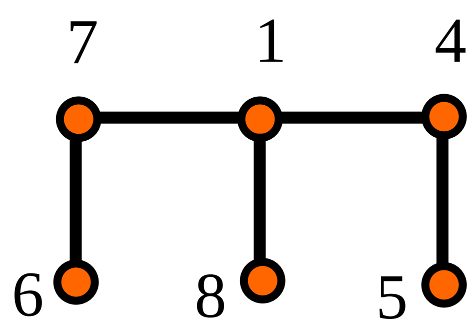
Full code (744171)

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY



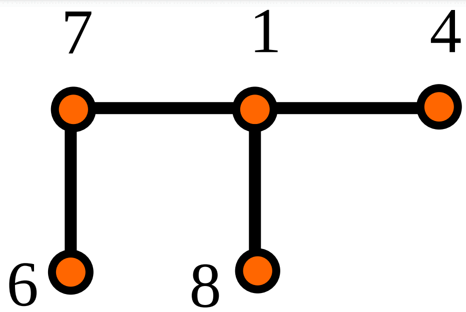


$a=(7)$

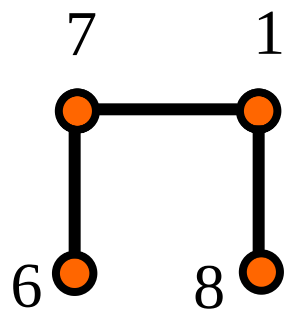


$a=(74)$

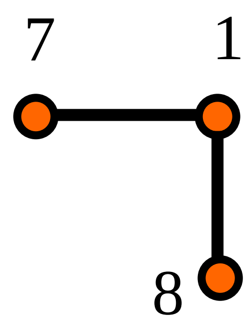
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY



$a=(744)$



$a=(7441)$

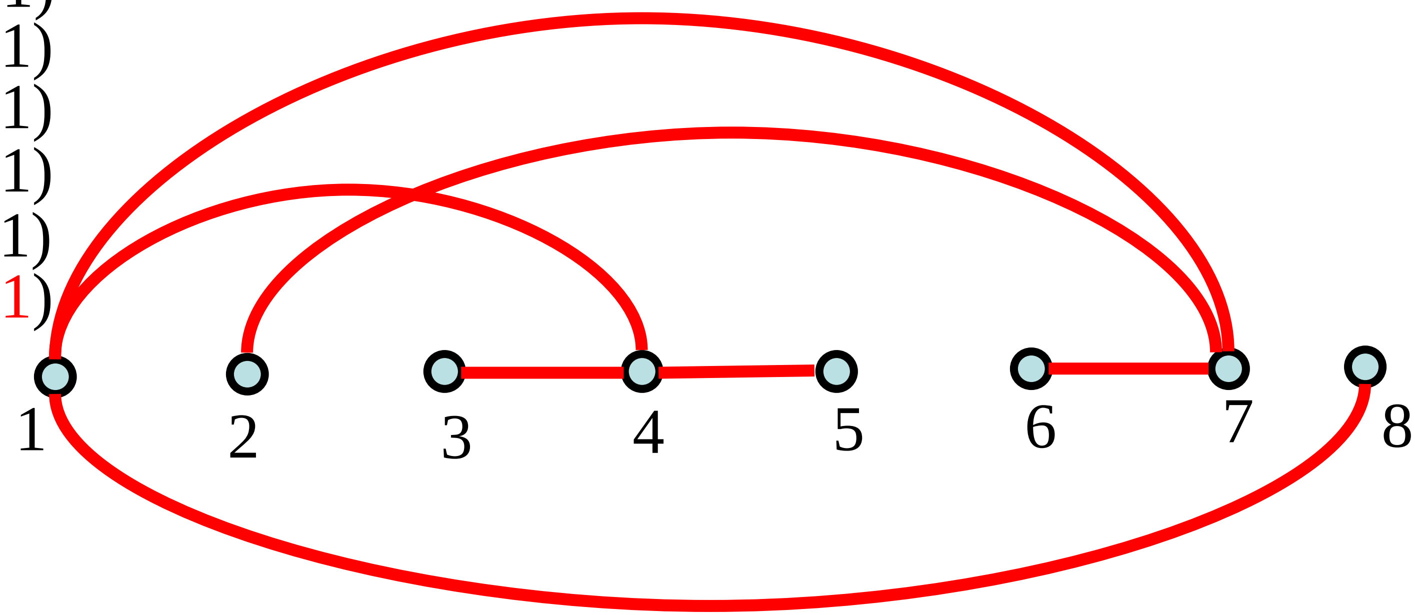


$a=(74417)$

$a=(744171)$

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

$a=(744171)$
$a=(744171)$
$a=(744171)$
$a=(744171)$
$a=(744171)$
$a=(744171)$



COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Theorem. (Cayley's Formula [1889]). For a set $S \subseteq N$ of size n, there are n^{n-2} trees with vertex set S.

Proof: Assume

- Algorithm 2.2.1 d efines a bijection f from the set of trees with vertex set S to the set S^{n-2} of lists of length n-2 from $S \Rightarrow$ Show for each $a=(a_1,a_2,\ldots,a_{n-2})\in S^{n-2}$ that exactly one tree with vertex set S with $f(T)=a$.
- Basis: n=2. The 2-vertex tree corresponding to a length-0 list.
  I. S.: For $a\in S^{n-2}$, find all solutions to $f(T)=a$. Every such a tree has x as its least leaf and has edge xa_1

  By I.H., there is exactly one tree T' having vertex set $S'=S-\{x\}$, and prufer code $a'=(a_2, a_3,\ldots,a_{n-2})$.
- Therefore, there is exactly one tree T with $f(T)=a$.

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Corollary.

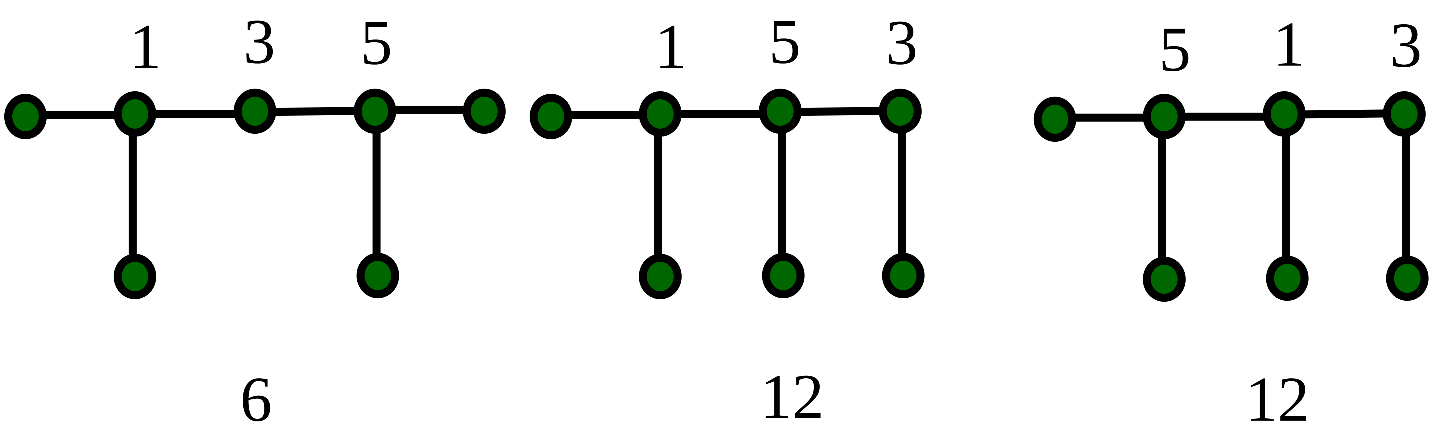
Given positive integers $d_1,\ldots,d_n$ summing to $2n$-2,there are exactly $\frac{(n-2)!}{\Pi(d_i-1)!}$ trees with vertex set $[n]$ such that vertex i has degree d_i, for each i.

Proof:

IDEA: each vertex x appears $d_T(x)-1$ times in the Prufer code.

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- Eg. Consider trees with vertices {1, 2, 3, 4, 5, 6, 7} that have degrees (3, 1, 2, 1, 3, 1, 1), respectively.
- Totally $\frac{(n-2)!}{\prod (d_i - 1)!} = 30$ trees.



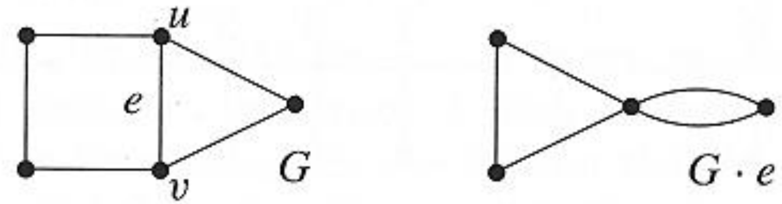
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Eg.

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Def:

In a graph G, contraction of edge e with endpoints u, v is the replacement of u and v with a single vertex whose incident edges are the edges other than e that were incident to u or v.

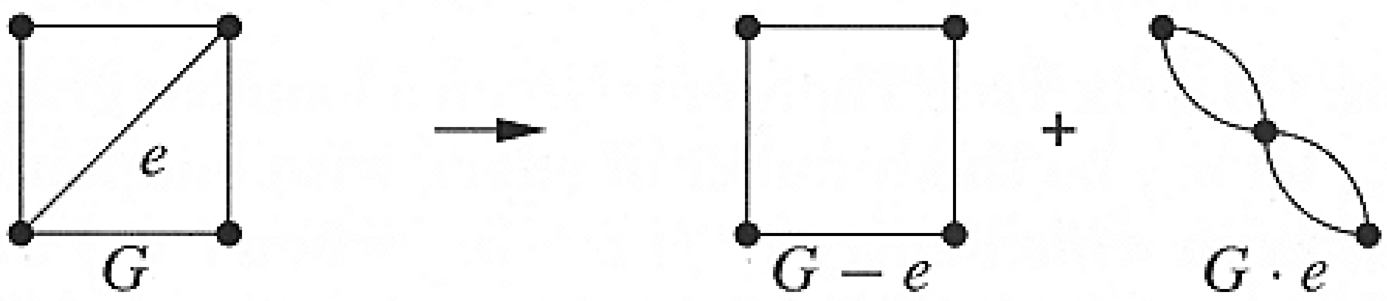


Proposition. Let $\tau(G)$ denote the number of spanning trees of a graph G. If $e \in E(G)$ is not a loop, then

$\tau(G) = \tau(G - e) + \tau(G \cdot e)$.

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- Eg.



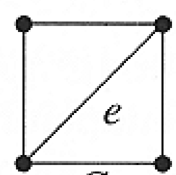
8=4+4

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- We cannot apply Proposition 2.2.8 when *e* is a loop!
- Eg. A graph consisting of one vertex and one loop.

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

2.2.12 Theorem. (Matrix Tree Theorem) Given a loopless graph G with vertex set $v_1,\ldots,v_n$, let $a_{i,j}$ be the number of edges with endpoints v_i and v_j. Let Q be the matrix in which entry (i, j) is $-a_{i,j}$ when $i \neq j$ and is $d(v_i)$ when $i = j$. If Q^* is a matrix obtained by deleting row s and column t of Q, then $\tau(G) = (-1)^{s+t} \det Q^*$.



G

$$\begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix} \rightarrow 8$$

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

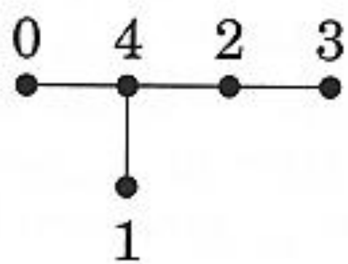
Def:

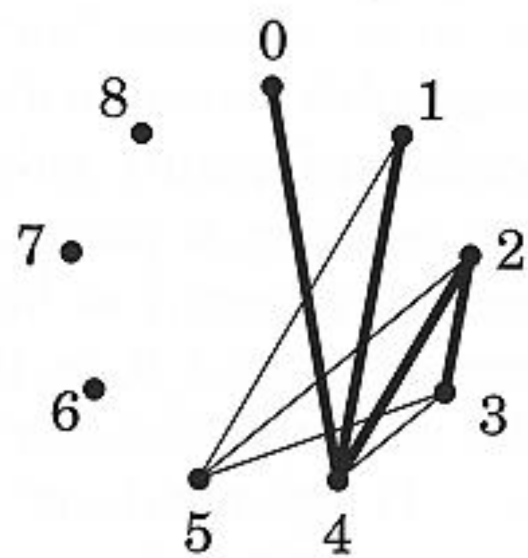
- A **graceful labeling** of a graph G with m edges is a function f : $V(G)\rightarrow\{0,\ldots, m\}$ such that distinct vertices receive distinct numbers and $\{|f(u)-f(v)| : uv \in E(G)\}=\{1,\ldots,m\}$

- A graph is **graceful** if it has a graceful labeling.

Theorem. If a tree T with m edges has a graceful labeling, then K_{2m+1} has a decomposition into $2m+1$ copies of T.

Idea:

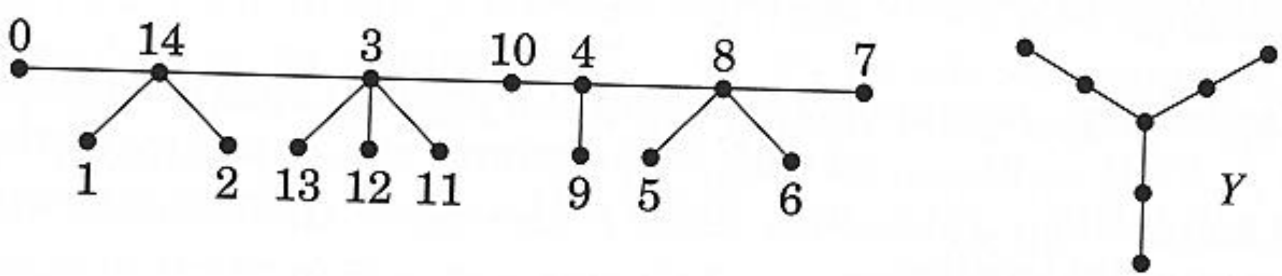




COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Def:

A *caterpillar* is a tree in which a simple path (the *spine*) is incident to (or contain) every edge.



Theorem. A tree is a caterpillar if and only if it does not contain the tree *Y* above.

Proof:

G' : obtained from G by deleting the leaves of G

G' has a vertex of degree at least 3 iff Y appears in G

$\therefore \Delta(G') \leq 2 \Leftrightarrow G'$ is a path.

Branching and Eulerian Digraphs

成功大學
COPYRIGHT 2002 NATIONAl CHENG KUNG UNIVERSITY

- A **branching** or **out-tree** is an orientation of a tree having a root of indegree 0 and all other vertices of indegree 1.
- An **in-tree** is an out-tree with edges reversed.

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Lemma. In a strong digraph, every vertex is the root of an out-tree (and an in-tree).

Proof: Consider a vertex v.

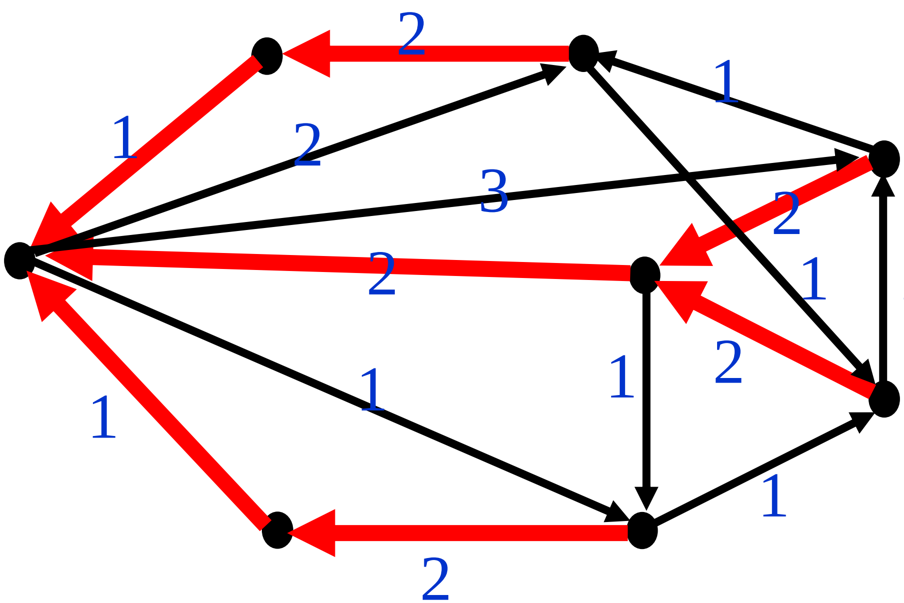
- Iteratively add edges to grow a branching from v.
- Let S_i=set of vertices reached i edges have been added. Initially, $S_0=\{v\}$.
- Since the digraph is strong $\Rightarrow \exists$ an edge leaving S_i add one such edge to the branching and add its head to S_i to obtain $S_{i+1} \Rightarrow$ until reaching all vertices
- Reverse all edges and apply the same procedure to obtain an in-tree

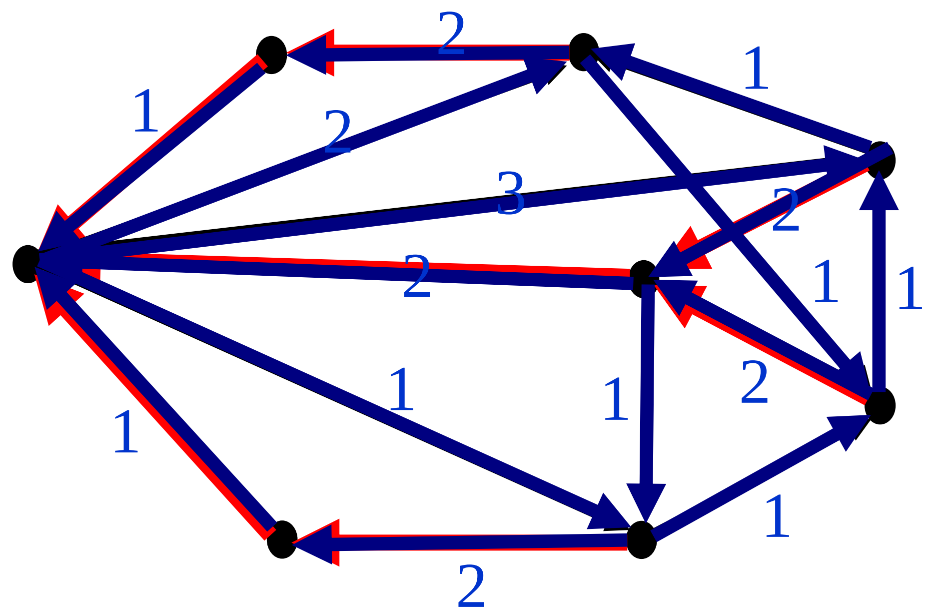
2.2.24 Algorithm (Eulerian circuit in directed graph)

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- Input: An Eulerian digraph G without isolated vertices and a spanning in-tree T consisting of paths to v
  - **S1**: For each $u \in V(G)$, specify an ordering of the edges that leave u, s.t. for $u \neq v$ the edges leaving u in T comes last.
  - **S2**: Beginning at v, construct an Eulerian circuit by always exiting the current vertex u along the next unused edge in the ordering specified at u

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY





Theorem. Algorithm 2.2.24 always produces an Eulerian circuit.

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Proof:

- Show the trail can end only at v after traversing all edges
- When we enter $u \neq v$, the edge leaving u in T has not yet been used ($d^+(u)=d^-(u)$) $\Rightarrow$ the trail can only end at v
- When we cannot continue $\Rightarrow$ stop at v and have used all exiting edges $\Rightarrow$ also used all edges entering v
- We cannot use an edge of T until it is only remaining edge leaving its tail $\Rightarrow$ cannot use all edges entering v until we have finished all the other vertices

  (T contains a path from each vertex to v)

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- Definition: Two Eulerian circuits are the same if the successive pairs of edges are the same.
- Theorem (van Aardenne-Ehrenfest and de Bruijn [1951]).

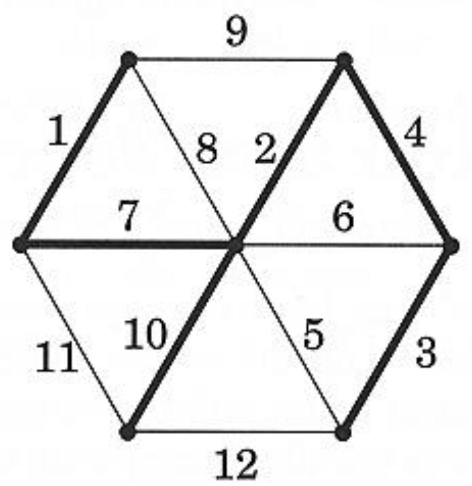
  In an Eulerian digraph with $d_i = d^+(v_i) = d^-(v_i)$ the number of Eulerian circuits is $c\Pi\,(d_i - 1)!$, where c counts the in-trees to or out-trees from any vertex.

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

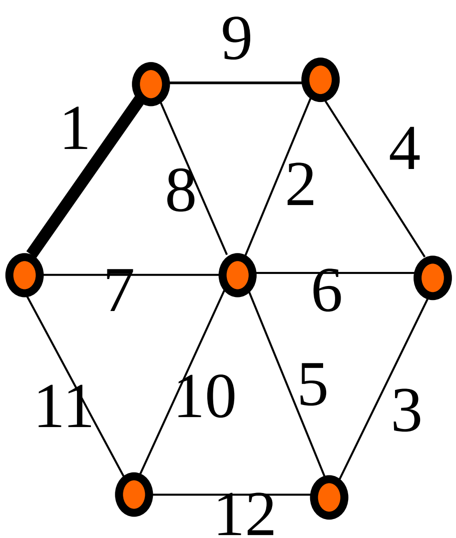
2.3.1.algorithm.(Kruskal's Algorithm – for minimum spanning trees.)

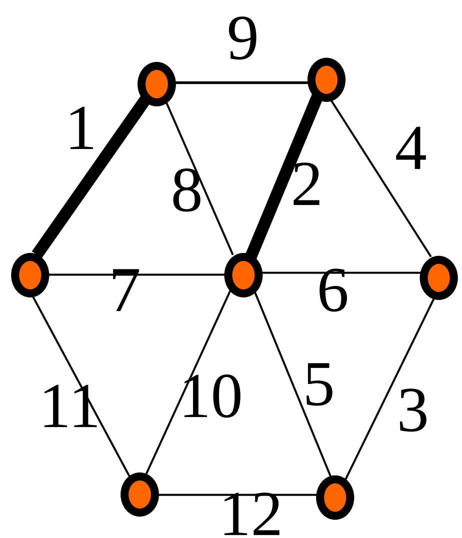
Input: A weighted connected graph.

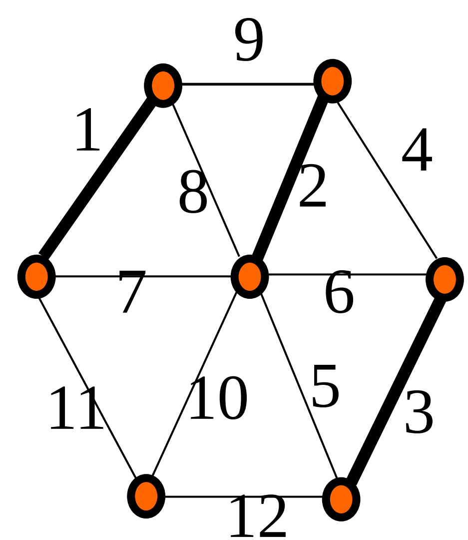
- Idea: Maintain an acyclic spanning subgraph H, enlarging it by edges with low weight to form a spanning tree. Consider edges in nondecreasing order of weight, breaking ties arbitrarily.
- Initialization: Set $E(H)=\varnothing$.
- Iteration: If the next cheapest edge joins two components of H, then include it; otherwise, discard it. Terminate when H is connected.



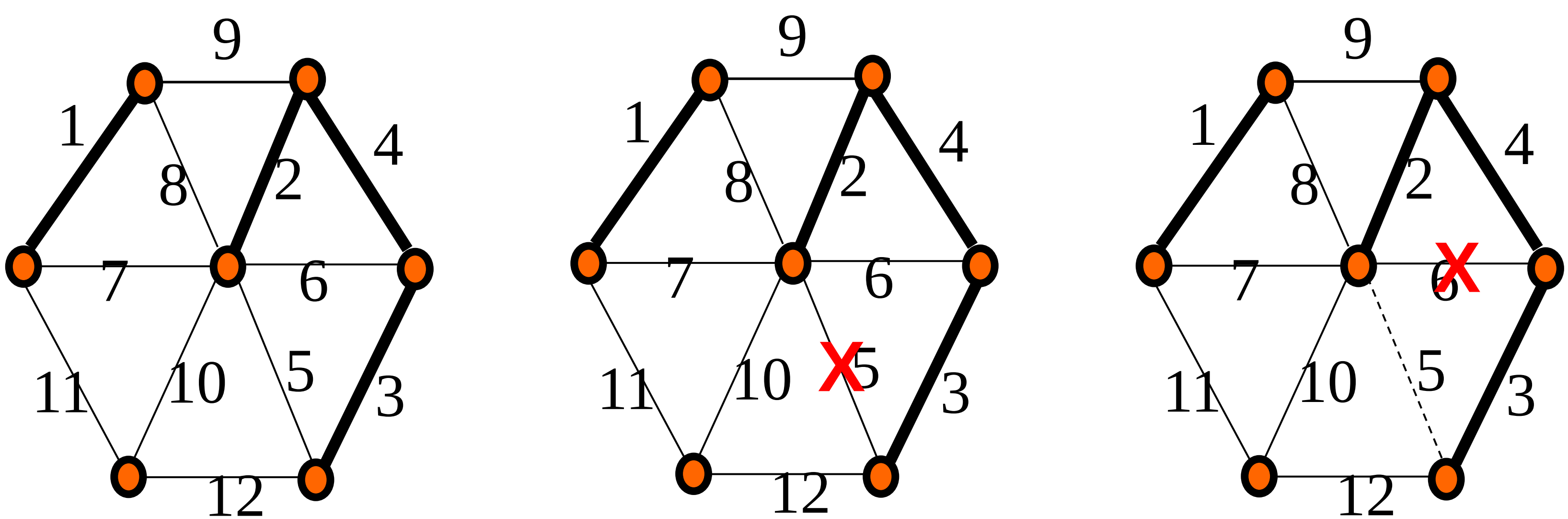
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY



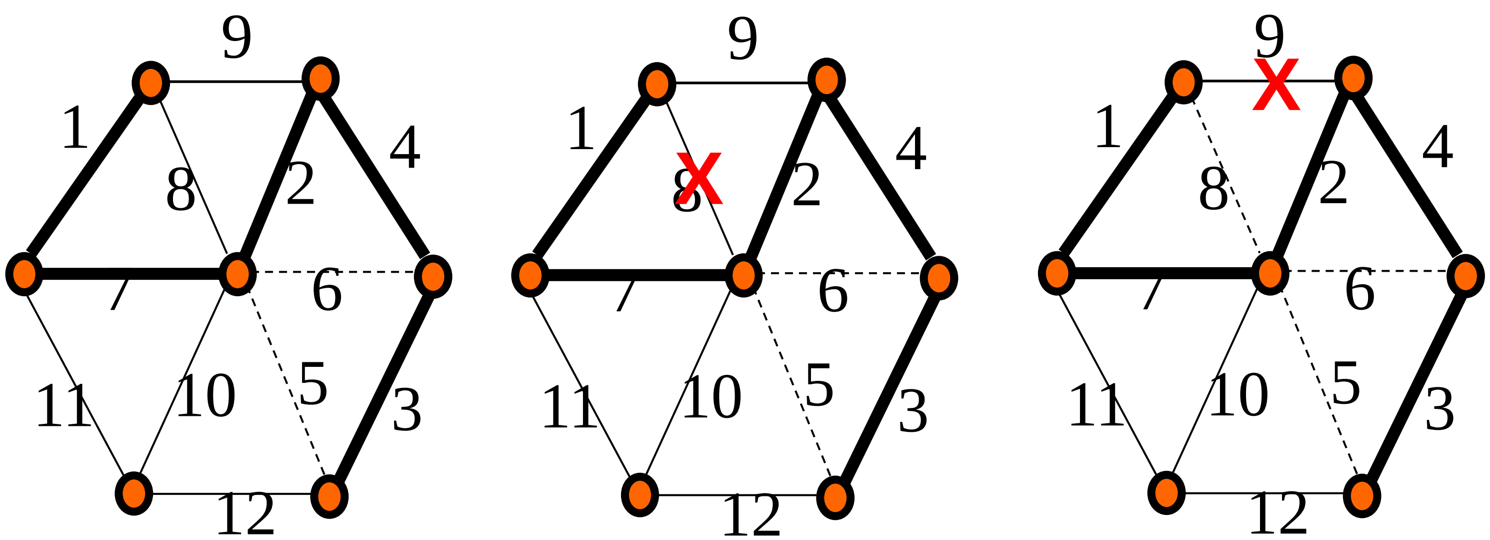


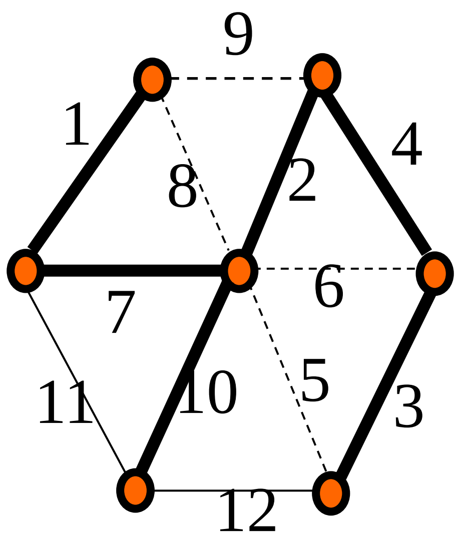


COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY



成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY




9
1
4
8
2
7
6
11
10
5
3
12

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Theorem. In a connected weighted graph G, Kruskal's Algorithm constructs a minimum-weight spanning tree.

Pf:

- The algorithm produces a connected and acyclic graph.
- T: the resulting tree produced by the algorithm
- T^*: a spanning tree of minimum weight
  - If $T=T^*$, we are done.
  - If $T\neq T^*$, let e be the first edge chosen for T that is not in T^*
    - T^*+e creates a cycle C and C has an edge $e' \notin E(T)$
    - $w(e)\leq w(e')$
    - $T^* +e- e'$ is a spanning tree with weight at most T^*
- Repeat the above yields a minimum-weight spanning tree that agrees completely with T.

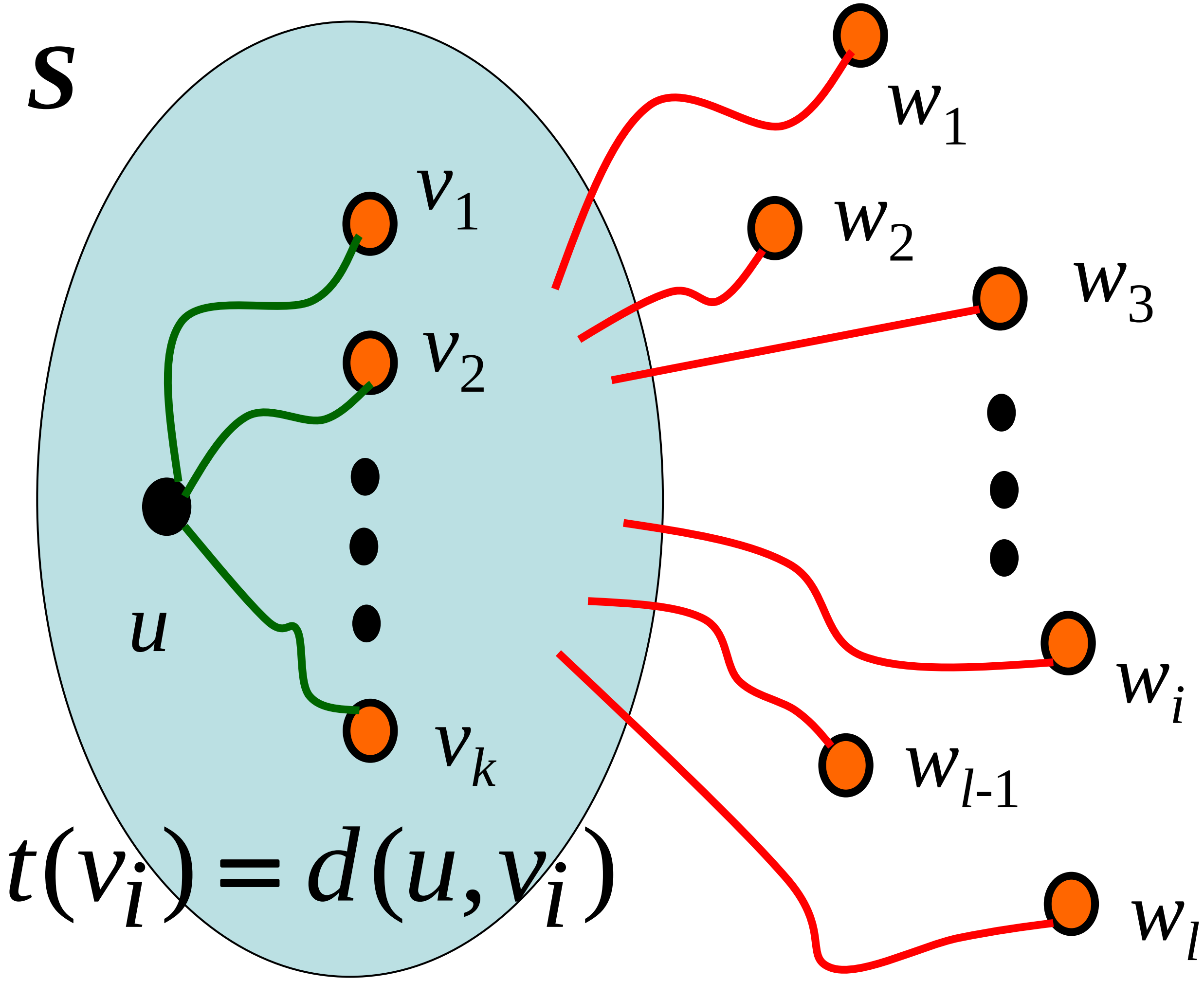
Dijkstra's Algorithm-distances from one vertex

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- Input: A graph (or digraph) with nonnegative edge weights and a starting vertex u. The weight of edge xy is $w(xy)$; let $w(xy) = \infty$ if xy is not an edge.
- Initialization: Set $S = \{u\}$; $t(u) = 0$; $t(z) = w(uz)$ for $z \neq u$.
- Iteration:
  - Select a vertex v outside S such that $t(v) = \min_{z \notin S} t(z)$.
  - Add v to S.
  - Explore edges from v to update tentative distance: for each edge vz with $z \notin S$, update $t(z)$ to $\min\{t(z), t(v) + w(vz)\}$.
- The iteration continues until $S=V(G)$ or until $t(z)=\infty$ for every $z \notin S$. At the end, set $d(u, v) = t(v)$ for all v.

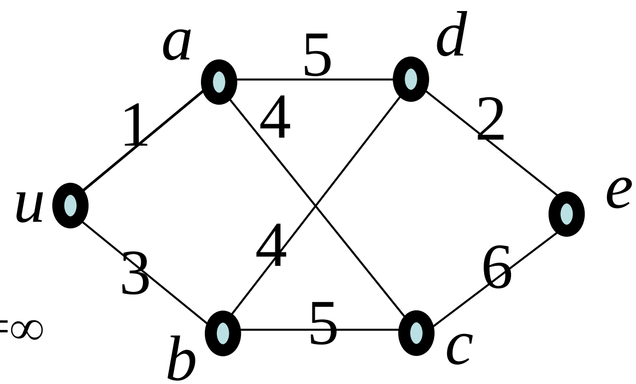
成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- $t(w_i)$ is the least length of a u,w_i-path reaching w_i from u

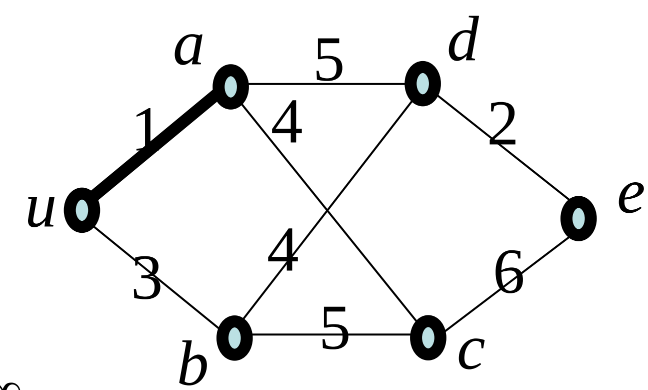


成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

$S=\{u\}$
$t(u)=0$
$t(a)=1$
$t(b)=3$
$t(c)=t(d)=t(e)=\infty$

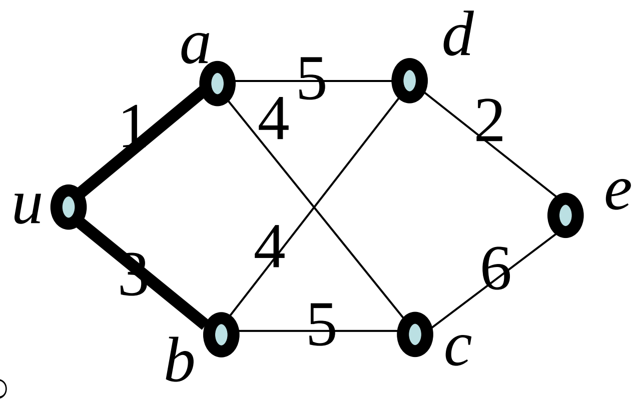


$S=\{u,a\}$
$t(u)=0$
$t(a)=1$
$t(b)=3$
$t(c)=5$
$t(d)=6 \quad t(e)=\infty$

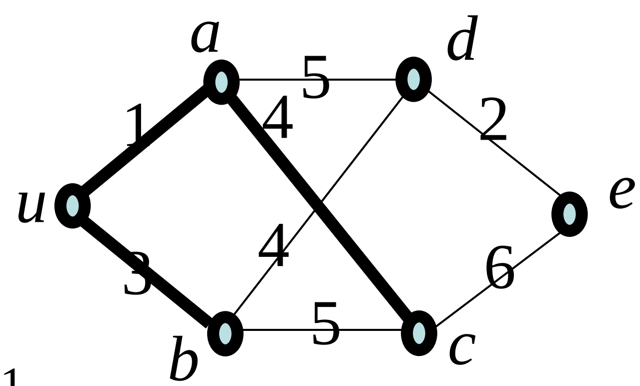


COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

$S=\{u,a,b\}$
$t(u)=0$
$t(a)=1$
$t(b)=3$
$t(c)=5$
$t(d)=6$ $t(e)=\infty$

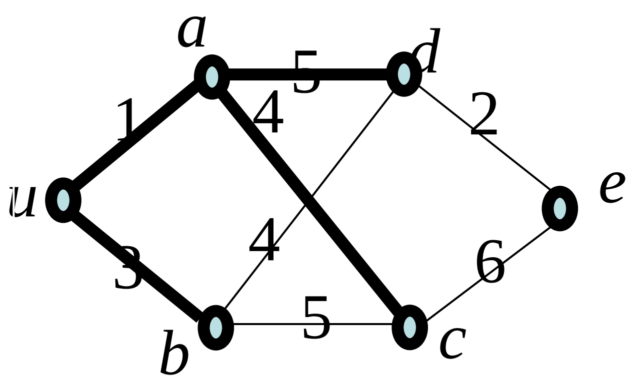


$S=\{u,a,b,c\}$
$t(u)=0$
$t(a)=1$
$t(b)=3$
$t(c)=5$
$t(d)=6$ $t(e)=11$

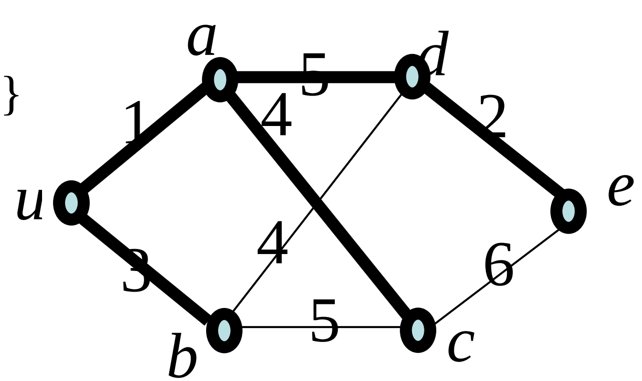


COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

$S=\{u,a,b,c,d\}$
$t(u)=0$
$t(a)=1$
$t(b)=3$
$t(c)=5$
$t(d)=6$ $t(e)=8$



$S=\{u,a,b,c,d,e\}$
$t(u)=0$
$t(a)=1$
$t(b)=3$
$t(c)=5$
$t(d)=6$ $t(e)=8$



COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

Theorem. Given a (di)graph G and a vertex Dijkstra's Algorithm computes $d(u,z)$ for every $z \in V(G)$

Pf:

By induction on $k=|S|$ to show

(1) For $z \in S$, $t(z) = d(u, z)$,

(2) For $z \notin S$, $t(z)$ is the least length of a u,z-path reaching z from S

- Basis on k=1.
- I.S.: Suppose that $|S|=k$, (1) and (2) are true.

  v: a vertex among $z \notin S$ such that $t(z)$ is smallest.

Let $S' = S \cup \{v\}$

- A shortest u,v-path must exit S before reaching v.
- By I.H., the length of the shortest path from S to v is $t(v)$.
- By I.H. and the choice of v, a path visiting any vertex outside S and later to v has length at least $t(v) \Rightarrow d(u, v) = t(v)$

To show (2):

z: a vertex outside *S* other then *v*

- By I.H., the shortest *u*,*z*-path reaching *z* directly from *S* has length $t(z)$;
- When *v* is added, the desired value for *z* is $\min\{t(z),\mathbf{t(v)+w(vz)}\}$

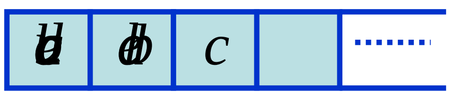
2.3.8 Algorithm (Breadth-First Search—BFS)

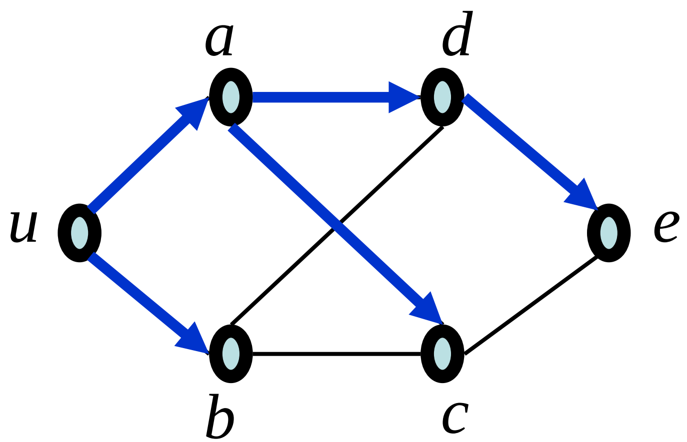
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- Input: An unweighted graph (or digraph) and a start vertex u.
- Idea: Maintain a set R of vertices that have been reached but not searched and a set S of vertices that have been searched. The set R is maintained as a First-In First-Out list (queue), so the first vertices found are the first vertices explored.
- Initialization: $R = \{u\}$, $S = \varnothing$, $d(u, u)=0$.
- Iteration: As long as $R \neq \varnothing$, we search from the first vertex v of R. The neighbors of v not in $S \cup R$ are added to the back of R and assigned distance $d(u, v)+1$, and then v is removed from the front R and placed in S.

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

R:





$S = \{u, a, b, d, c, e\}$

$d(u, e) = 3$

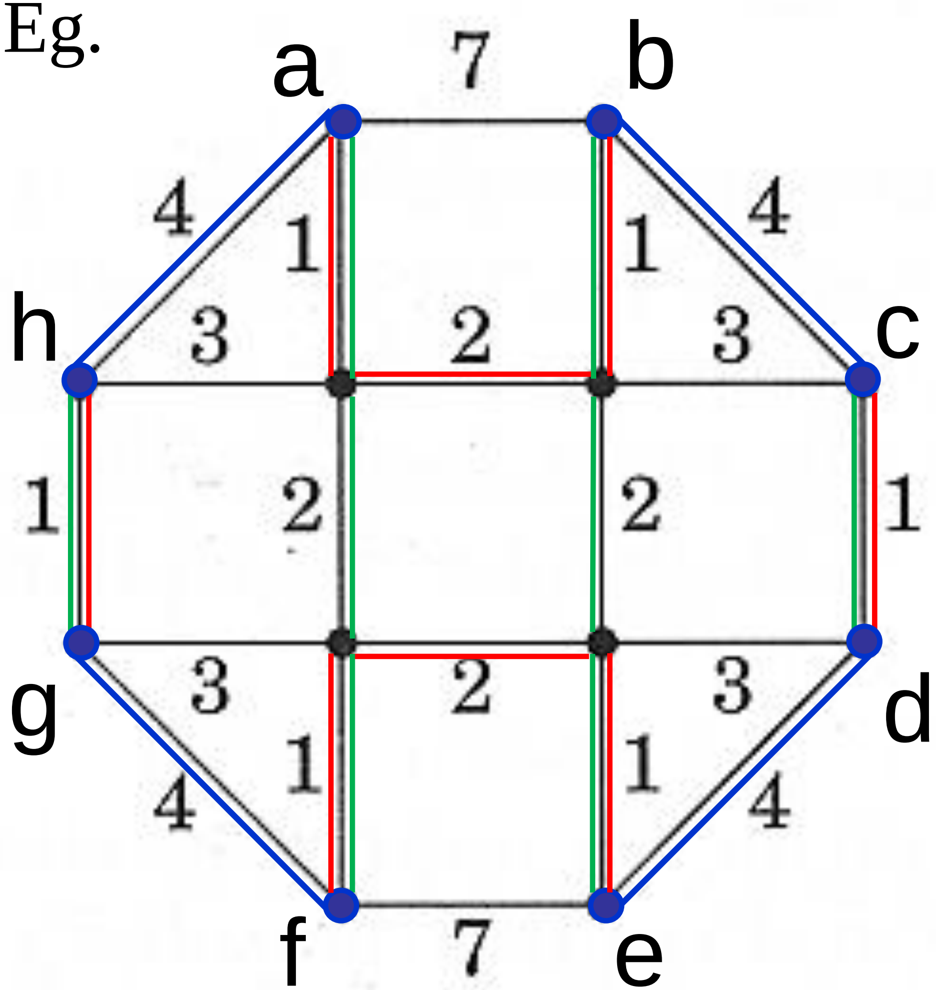
$d(u, b) = 1$

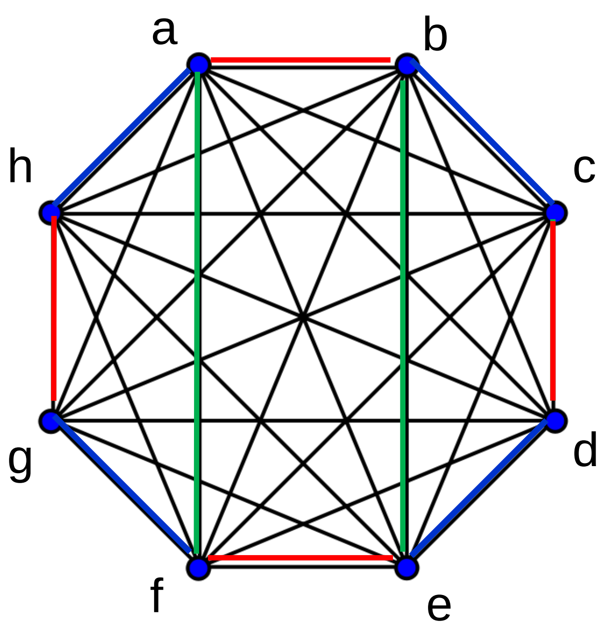
Chinese Postman Problem:

COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- A mail carrier must traverse all edges in a road network, starting and ending at the Post Office.
- The edges have nonnegative weights representing distance or time.
- We seek a closed walk of minimum total length that uses all the edges.
- Key:
  - If every vertex is even, then the graph is Eulerian and the answer is the sum of the edge weights.
  - Otherwise, we must repeat edges.

- Eg.





4+4+4+4=16

1+1+2+1+1+2+1+1=10

1+2+1+1+2+1+1+1=10

成功大學
COPYRIGHT 2002 NATIONAL CHENG KUNG UNIVERSITY

- If there are only two odd vertices, then use Dijkstra's algorithm to find the shortest path between them.

- If there are $2k$ odd vertices, then
  - Use Dijkstra's algorithm to find the shortest paths connecting each pair of odd vertices;
  - Use these lengths as weights on the edges of K_{2k};
  - Solve the weighted version of the maximum matching problem on K_{2k}.