Lecture 4: Trees (2) and Circuits

Shuai Li

John Hopcroft Center, Shanghai Jiao Tong University

https://shuaili8.github.io

https://shuaili8.github.io/Teaching/CS445/index.html

The center of a tree

- Theorem (1.15, H) In any tree, the center is either a single vertex or a pair of adjacent vertices

Tree as subgraphs

- Theorem (1.16, H) Let T be a tree of order $k + 1$ with k edges. Let G be a graph with $\delta(G) \geq k$. Then G contains T as a subgraph

Spanning tree

- Given a graph G and a subgraph T, T is a spanning tree of G if T is a tree that contains every vertex of G
- Example: A telecommunications company tries to lay cable in a new neighbourhood
- Proposition (2.1.5c, W) Every connected graph contains a spanning tree

Minimal spanning tree - Kruskal's Algorithm

- Given: A connected, weighted graph G

1. Find an edge of minimum weight and mark it.
2. Among all of the unmarked edges that do not form a cycle with any of the marked edges, choose an edge of minimum weight and mark it
3. If the set of marked edges forms a spanning tree of G, then stop. If not, repeat step 2

Example

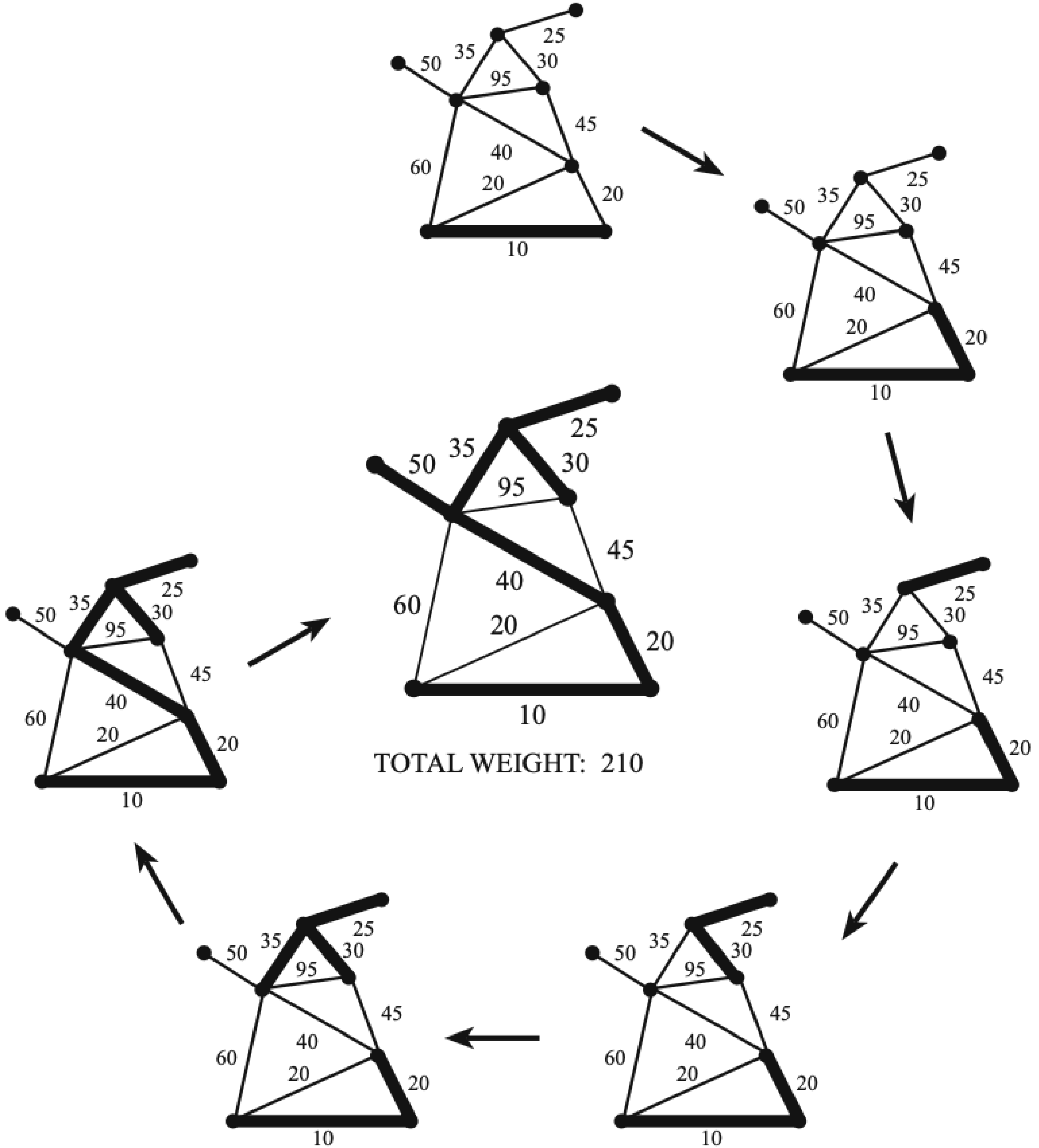


FIGURE 1.43. The stages of Kruskal's algorithm.

Theoretical guarantee of Kruskal’s algorithm

- Theorem (1.17, H) Kruskal’s algorithm produces a spanning tree of minimum total weight

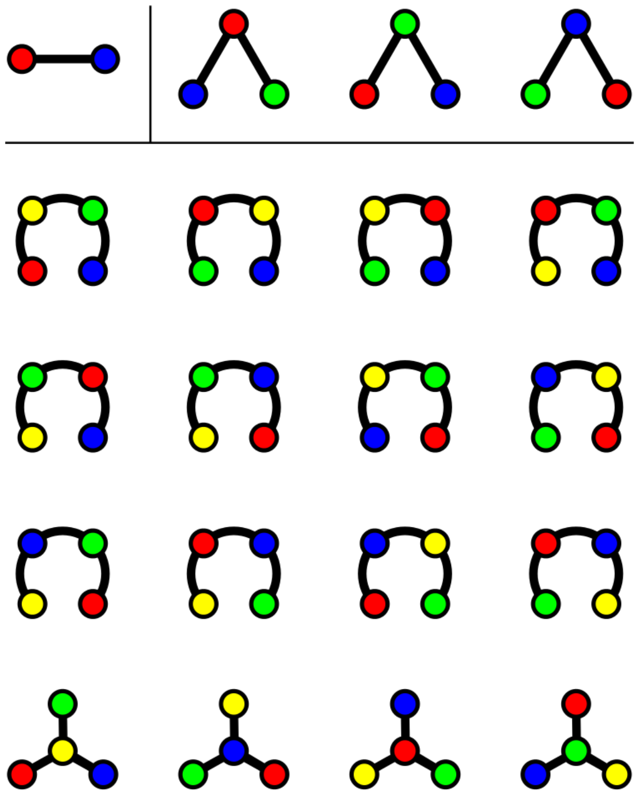
Prim’s Algorithm

- Given: A connected, weighted graph G.

1. Choose a vertex v, and mark it.
2. From among all edges that have one marked end vertex and one unmarked end vertex, choose an edge e of minimum weight. Mark the edge e, and also mark its unmarked end vertex.
3. If every vertex of G is marked, then the set of marked edges forms a minimum weight spanning tree. If not, repeat step 2

Cayley’s tree formula

- Theorem (1.18, H). There are n^{n-2} distinct labeled trees of order n



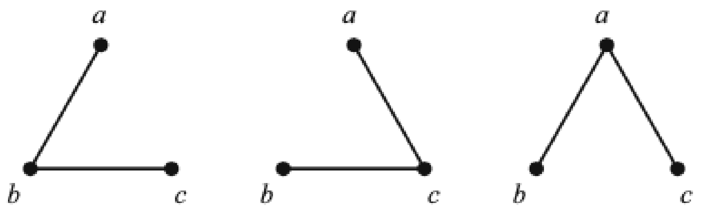


FIGURE 1.45. Labeled trees on three vertices.

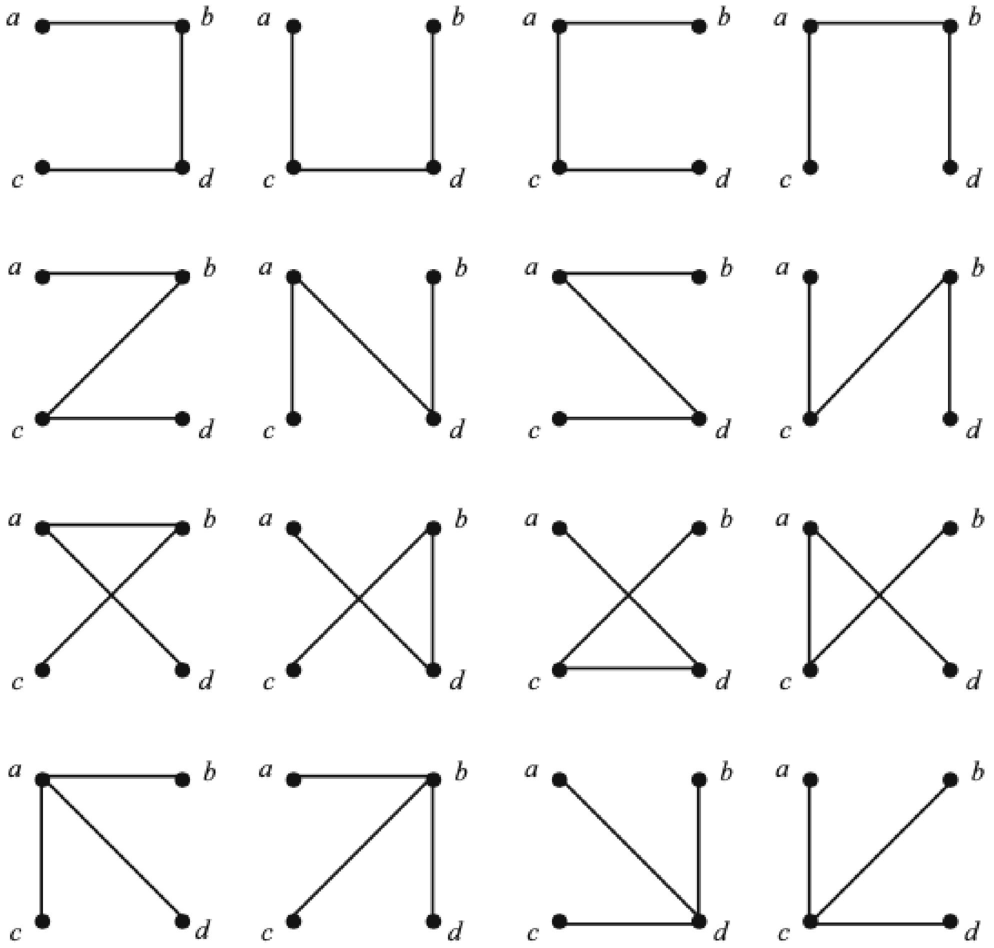


FIGURE 1.46. Labeled trees on four vertices.

Example

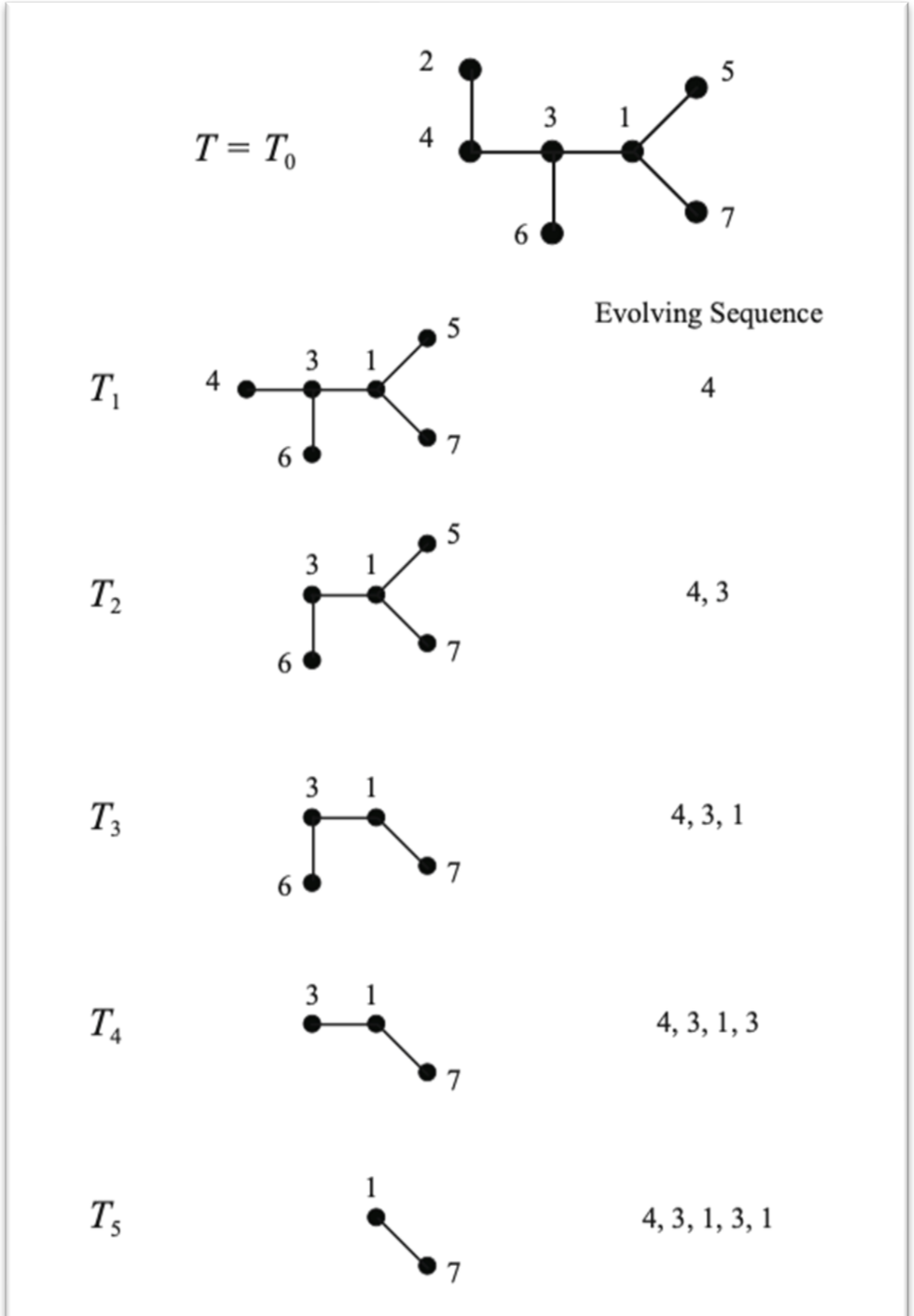


FIGURE 1.47. Creating a Prüfer sequence.

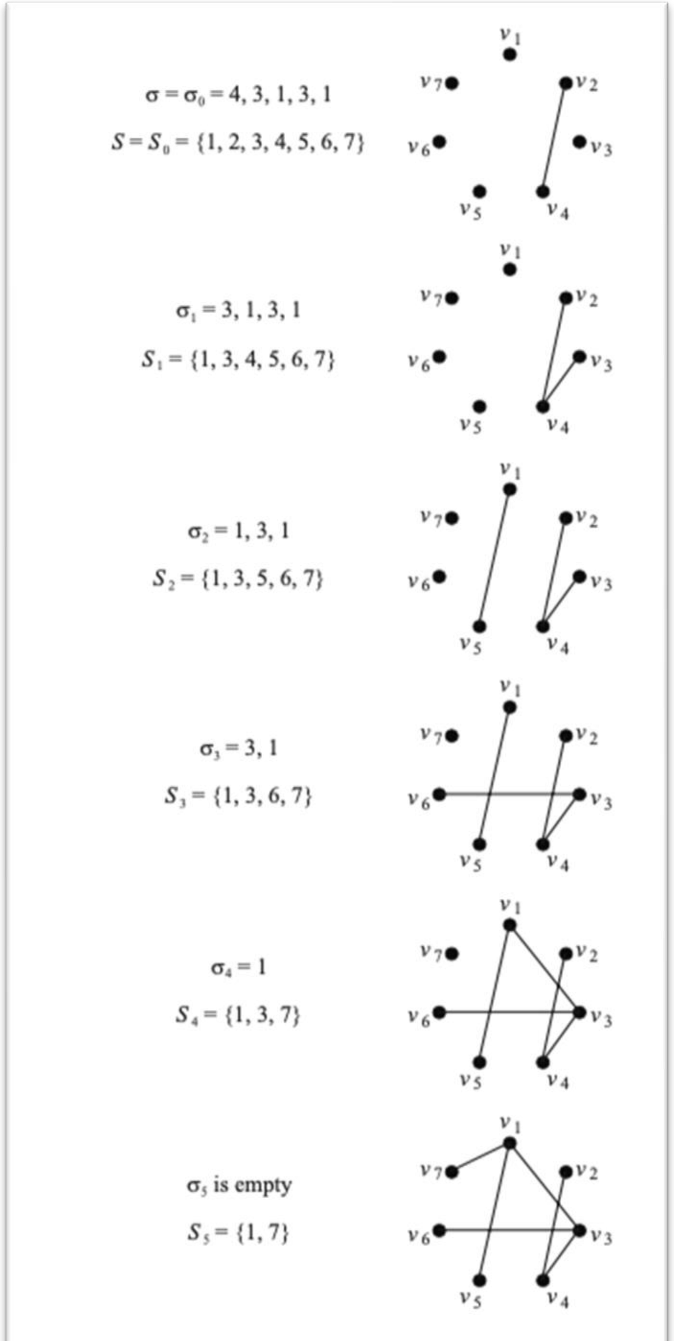
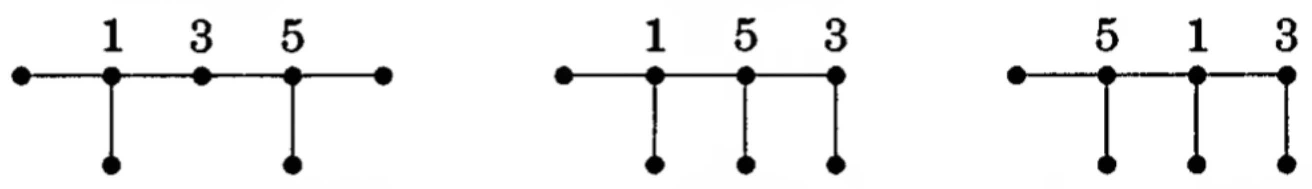


FIGURE 1.48. Building a labeled tree.

Trees with fixed degrees

- Corollary (2.2.4, W) Given positive integers $d_1, \ldots, d_n$ summing to $2n-2$, there are exactly $\frac{(n-2)!}{\prod(d_i-1)!}$ trees with vertex set $[n]$ such that vertex i has degree d_i for each i

- Example (2.2.5, W) Consider trees with vertices $[7]$ that have degrees $(3,1,2,1,3,1,1)$



Matrix tree theorem - cofactor

- For an $n \times n$ matrix A, the i, j cofactor of A is defined to be
$$(-1)^{i+j} \det(M_{ij})$$
where M_{ij} represents the $(n-1) \times (n-1)$ matrix formed by deleting row i and column j from A

3 × 3 generic matrix [edit]

Consider a 3×3 matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Its cofactor matrix is

$$\mathbf{C} = \begin{pmatrix} +\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} \\ +\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & +\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix},$$

Matrix tree theorem

- Theorem (1.19, H; 2.2.12, W; Kirchhoff) If G is a connected labeled graph with adjacency matrix A and degree matrix D, then the number of unique spanning trees of G is equal to the value of any cofactor of the matrix $D - A$
- If the row sums and column sums of a matrix are all 0, then the cofactors all have the same value
- Exercise Read the proof part
- Exercise (Ex7, S1.3.4, H) Use the matrix tree theorem to prove Cayley's theorem

Example

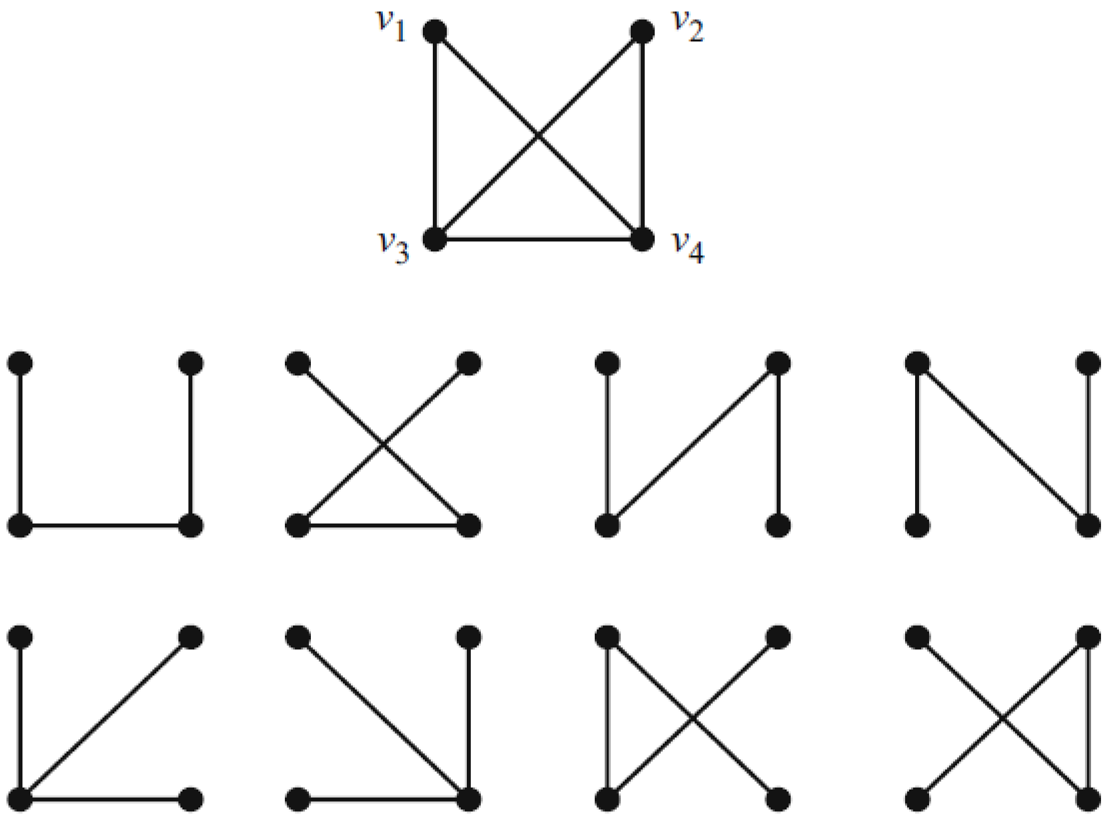


FIGURE 1.49. A labeled graph and its spanning trees.

The degree matrix D and adjacency matrix A are

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix},$$

and so

$$D - A = \begin{bmatrix} 2 & 0 & -1 & -1 \\ 0 & 2 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}.$$

The $(1,1)$ cofactor of $D - A$ is

$$\det \begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix} = 8.$$

Score one for Kirchhoff!

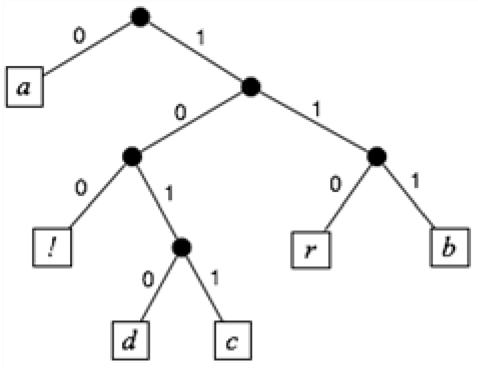
Wiener index

- In a communication network, large diameter may be acceptable if most pairs can communicate via short paths. This leads us to study the average distance instead of the maximum
- Wiener index $D(G) = \sum_{u,v \in V(G)} d_G(u, v)$
- Theorem (2.1.14, W) Among trees with n vertices, the Wiener index $D(T)$ is minimized by stars and maximized by paths, both uniquely

Prefix coding

- A binary tree is a rooted plane tree where each vertex has at most two children
- Given large computer files and limited storage, we want to encode characters as binary lists to minimize total length
- Prefix coding: no code word is an initial portion of another

- Example: 11001111011



character	encoding
a	0
b	111
c	1011
d	1010
r	110
!	100

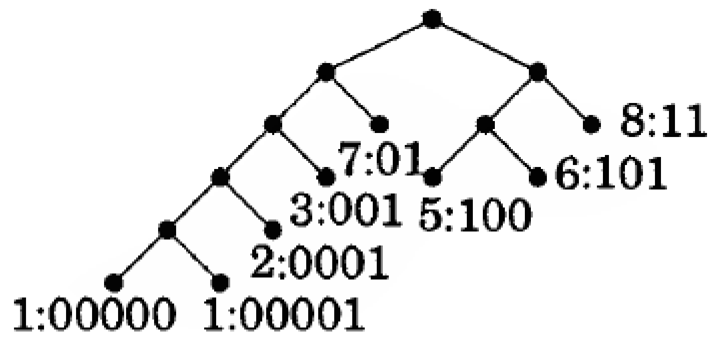
A binary prefix code for the alphabet {*a*, *b*, *c*, *d*, *r*, *!*}

Huffman coding

- Input: Weights (frequencies or probabilities) $p_1, \ldots, p_n$
- Output: Prefix-free code (equivalently, a binary tree)
- Idea: Infrequent items should have longer codes; put infrequent items deeper by combining them into parent nodes.
- Recursion: replace the two least likely items with probabilities p, p' with a single item of weight $p + p'$

Example (2.3.14, W)

a	5	100
b	1	00000
c	1	00001
d	7	01
e	8	11
f	2	0001
g	3	001
h	6	101



The average length is $\frac{5\times3+5+5+7\times2+\cdots}{33}=\frac{30}{11}<3$

Huffman coding is optimal

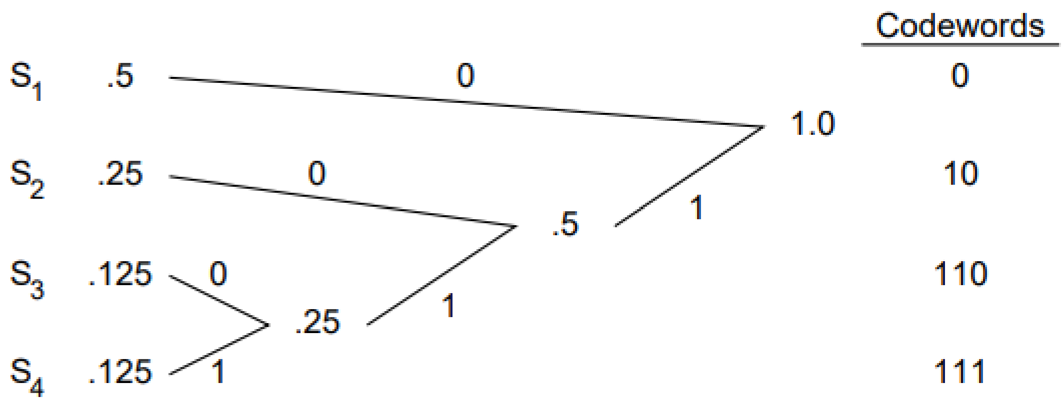
- Theorem (2.3.15, W) Given a probability distribution $\{p_i\}$ on n items, Huffman's Algorithm produces the prefix-free code with minimum expected length

Huffman coding and entropy

- The entropy of a discrete probability distribution $\{p_i\}$ is that

$$H(p) = -\sum_i p_i \log_2 p_i$$

- $H(p) \leq$ average length of Huffman coding $\leq H + 1$
- When each p_i is a power of ½, average length of Huffman coding is $H(p)$



$$\begin{aligned}\text{average length} &= (1)\left(\frac{1}{2}\right) + (2)\left(\frac{1}{4}\right) + (3)\left(\frac{1}{8}\right) + (3)\left(\frac{1}{8}\right) \\ &= 1.75 \ \text{ bits/symbol}\end{aligned}$$

$$\begin{aligned}H &= \frac{1}{2}\log_2 2 + \frac{1}{4}\log_2 4 + \frac{1}{8}\log_2 8 + \frac{1}{8}\log_2 8 \\ &= \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{3}{8} \\ &= 1.75\end{aligned}$$

Circuits

Eulerian circuit

- A closed walk through a graph using every edge once is called an Eulerian circuit
- A graph that has such a walk is called an Eulerian graph

- Theorem (1.2.26, W) A graph G is Eulerian $\Leftrightarrow$ it has at most one nontrivial component and its vertices all have even degree
- (possibly with multiple edges)
- Proof "$\Longrightarrow$" That G must be connected is obvious.
  Since the path enters a vertex through some edge and leaves by another edge, it is clear that all degrees must be even

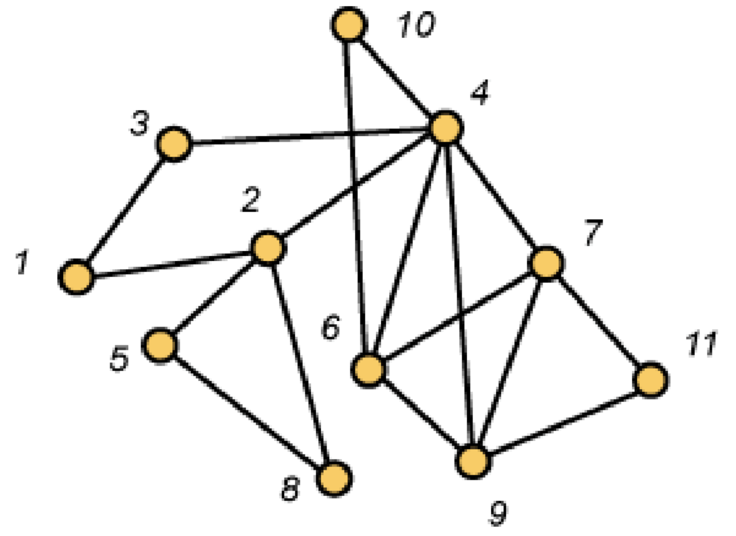
Key lemma

- Lemma (1.2.25, W) If every vertex of a graph G has degree at least 2, then G contains a cycle.

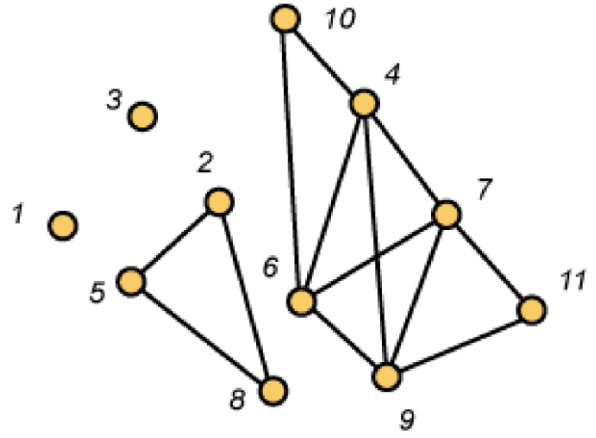
Algorithm for Euler Circuits

1. Choose a root vertex r and start with the trivial partial circuit (r)
2. Given a partial circuit $(x_0, e_1, x_1, \dots, x_{t-1}, e_t, x_t = x_0)$ that traverses not all edges of G, remove these edges from G
3. Let i be the least integer for which x_i is incident with one of the remaining edges
4. Form a greedy partial circuit among the remaining edges of the form $(x_i = y_0, e_1', y_1, \dots, y_{s-1}, e_s', y_s = x_i)$
5. Expand the original circuit by setting $(x_0, e_1, \dots, e_i, x_i = y_0, e_1', y_1, \dots, y_{s-1}, e_s', y_s = x_i, e_{i+1}, \dots, e_t, x_t = x_0)$
6. Repeat step 2-5

Example

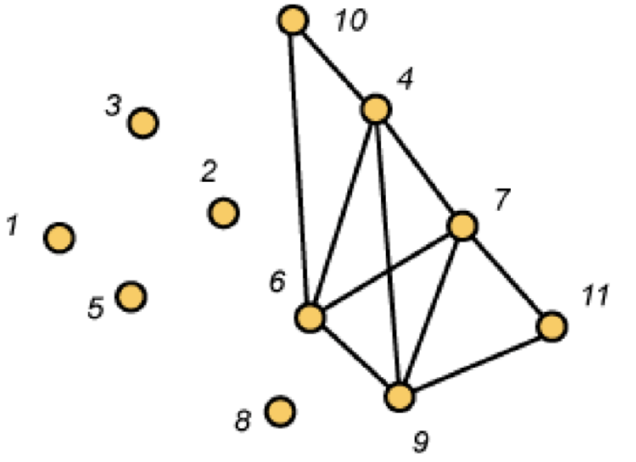


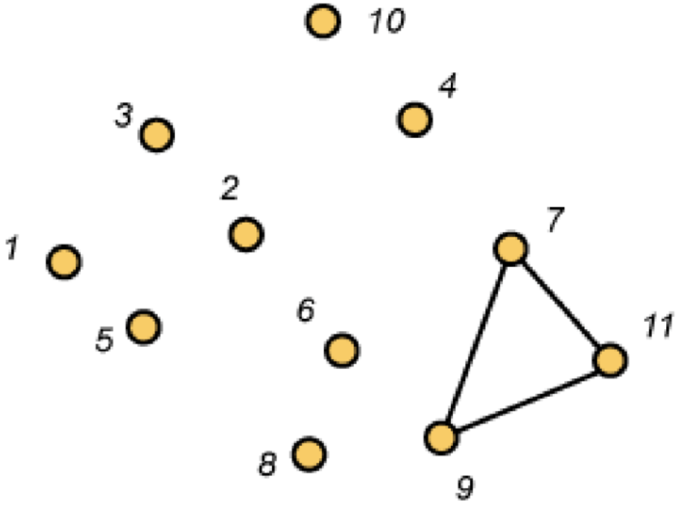
1. Start with the trivial circuit (1)
2. Greedy algorithm yields the partial circuit (1,2,4,3,1)
3. Remove these edges
4. The first vertex incident with remaining edges is 2
5. Greedy algorithms yields (2,5,8,2)
6. Expanding (1,2,5,8,2,4,3,1)
7. Remove these edges



Example (cont.)

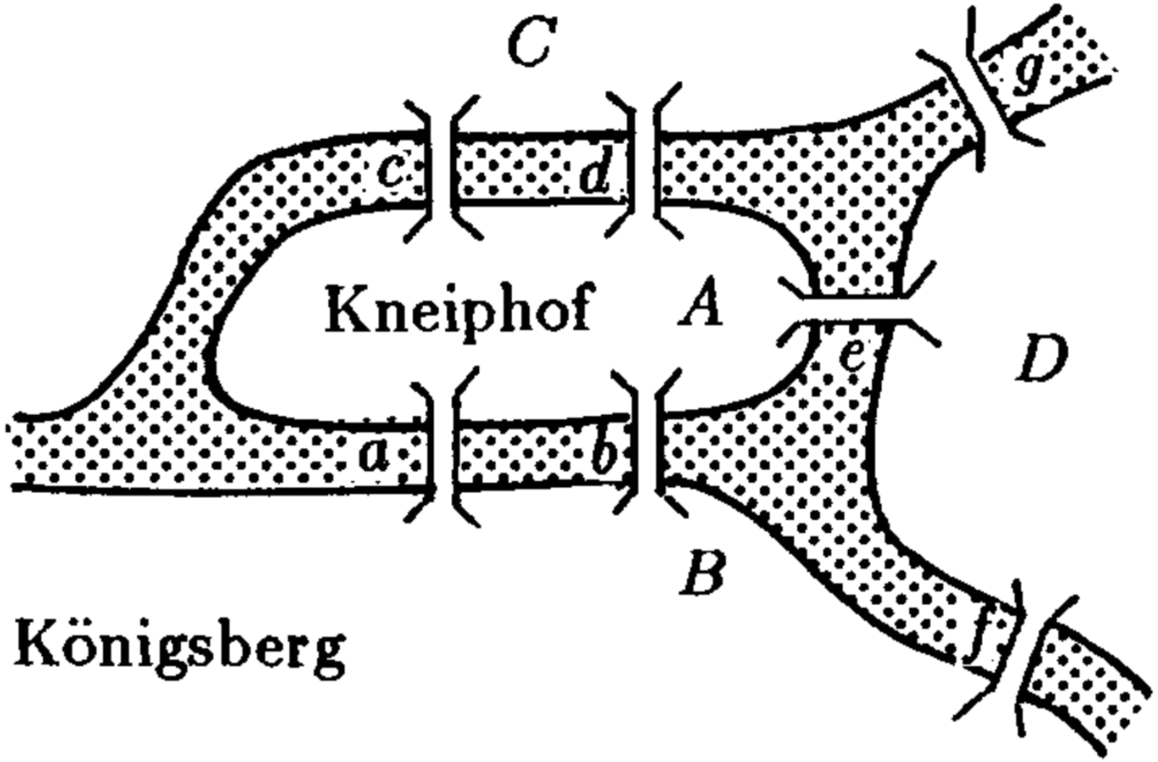
6. Expanding (1,2,5,8,2,4,3,1)
7. Remove these edges
8. First vertex incident with remaining edges is 4
9. Greedy algorithm yields (4,6,7,4,9,6,10,4)
10. Expanding (1,2,5,8,2,4,6,7,4,9,6,10,4,3,1)
11. Remove these edges
12. First vertex incident with remaining edges is 7
13. Greedy algorithm yields (7,9,11,7)
14. Expanding (1,2,5,8,2,4,6,7,9,11,7,4,9,6,10,4,3,1)

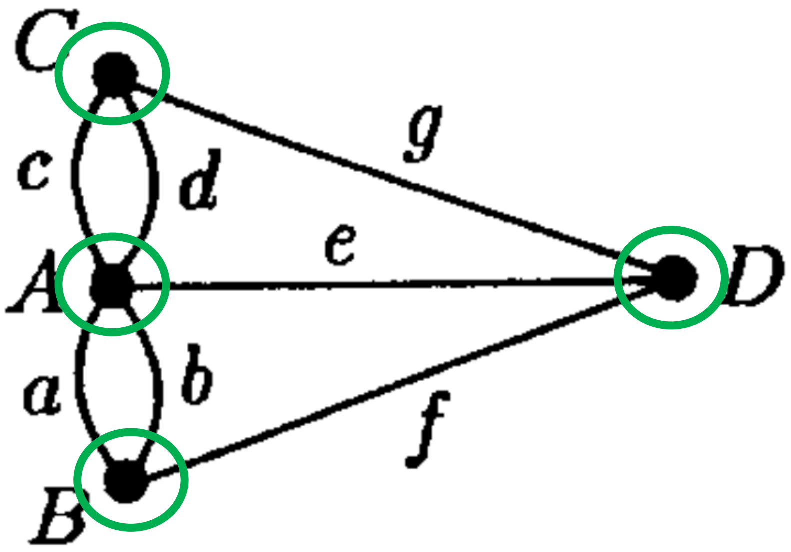




Eulerian circuit

- Theorem (1.2.26, W) A graph G is Eulerian $\Leftrightarrow$ it has at most one nontrivial component and its vertices all have even degree





Other properties

- Proposition (1.2.27, W) Every even graph decomposes into cycles

- The necessary and sufficient condition for a directed Eulerian circuit is that the graph is connected and that each vertex has the same ‘in-degree’ as ‘out-degree’

TONCAS

- TONCAS: The obvious necessary condition is also sufficient
- Theorem (1.2.26, W) A graph G is Eulerian $\Leftrightarrow$ it has at most one nontrivial component and its vertices all have even degree

- Proposition (1.3.28, W) The nonnegative integers $d_1, \ldots, d_n$ are the vertex degrees of some graph $\Leftrightarrow \sum_{i=1}^{n} d_i$ is even
- (Possibly with loops)
- Otherwise $(2,0,0)$ is not realizable

1.3.63. (!) Let $d_1, \ldots, d_n$ be integers such that $d_1 \geq \cdots \geq d_n \geq 0$. Prove that there is a loopless graph (multiple edges allowed) with degree sequence $d_1, \ldots, d_n$ if and only if $\sum d_i$ is even and $d_1 \leq d_2 + \cdots + d_n$. (Hakimi [1962])