Lecture 3: Bipartite Graphs and Trees

Shuai Li

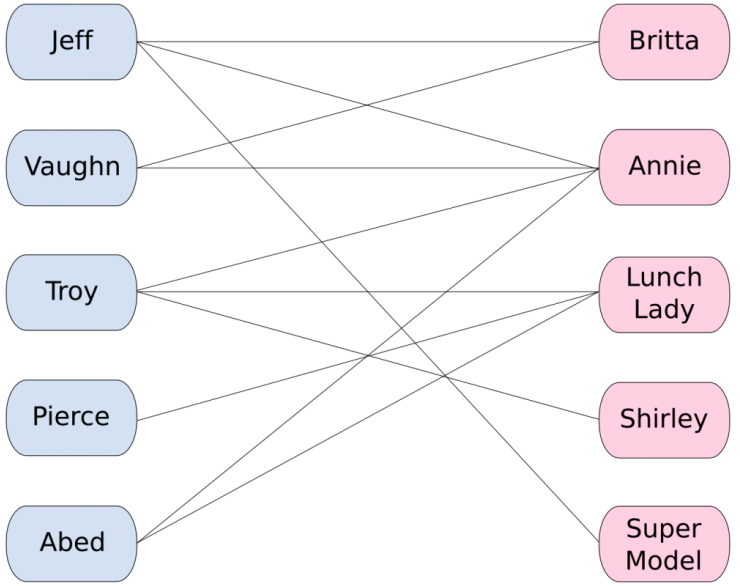
John Hopcroft Center, Shanghai Jiao Tong University

https://shuaili8.github.io

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

Bipartite graphs

- Theorem (1.2.18, W, Kőnig 1936)
  A graph is bipartite ⟺ it contains no odd cycle

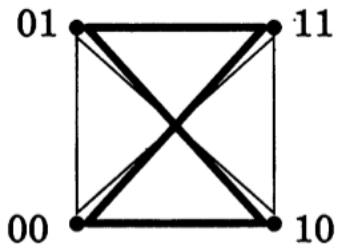


Proposition (1.2.15, W) Every closed odd walk contains an odd cycle

Complete graph is a union of bipartite graphs

- The union of graphs $G_1, \ldots, G_k$, written $G_1 \cup \cdots \cup G_k$, is the graph with vertex set $\bigcup_{i=1}^{k} V(G_i)$ and edge set $\bigcup_{i=1}^{k} E(G_i)$

- Consider an air traffic system with k airlines
  - Each pair of cities has direct service from at least one airline
  - No airline can schedule a cycle through an odd number of cities
  - Then, what is the maximum number of cities in the system?



- Theorem (1.2.23, W) The complete graph K_n can be expressed as the union of k bipartite graphs $\Leftrightarrow n \leq 2^k$

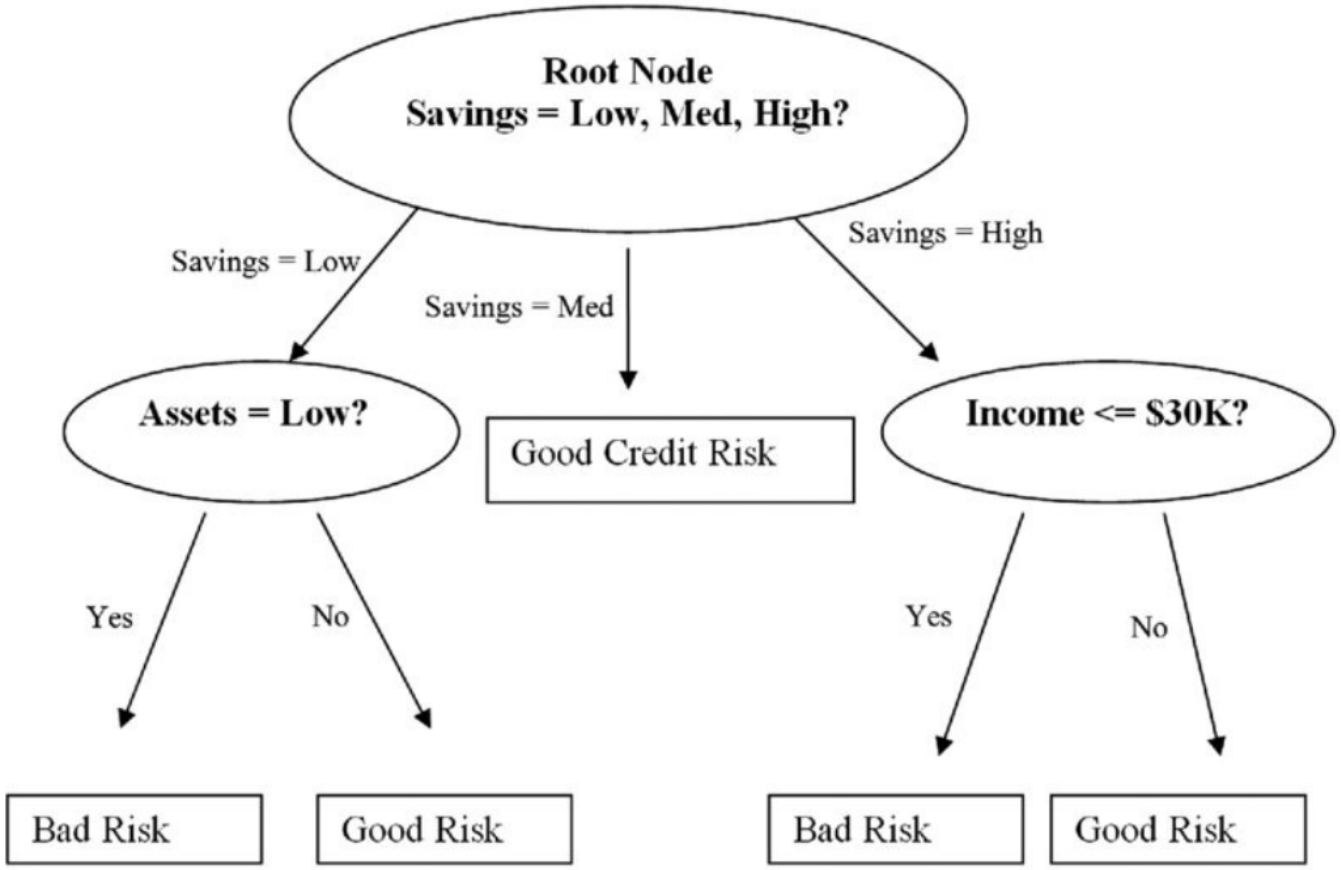
Bipartite subgraph is large

- Theorem (1.3.19, W) Every loopless graph G has a bipartite subgraph with at least $\lceil |E|/2 \rceil$ edges

Trees

Trees

- A tree is a connected graph T with no cycles



Properties

- Recall that a graph is bipartite $\Leftrightarrow$ it has no odd cycle
- (Ex 3, S1.3.1, H) A tree of order $n \geq 2$ is a bipartite graph

- Recall that an edge e is a bridge $\Leftrightarrow$ e lies on no cycle of G
- $\Rightarrow$ Every edge in a tree is a bridge
- T is a tree $\Leftrightarrow$ T is minimally connected, i.e. T is connected but $T - e$ is disconnected for every edge $e \in T$

Equivalent definitions (Theorem 1.5.1, D)

- T is a tree of order n
  $\Leftrightarrow$ Any two vertices of T are linked by a unique path in T
  $\Leftrightarrow$ T is minimally connected
  - i.e. T is connected but $T - e$ is disconnected for every edge $e \in T$

  $\Leftrightarrow$ T is maximally acyclic
  - i.e. T contains no cycle but $T + xy$ does for any non-adjacent vertices $x, y \in T$

  $\Leftrightarrow$ (Theorem 1.10, 1.12, H) T is connected with $n - 1$ edges

  $\Leftrightarrow$ (Theorem 1.13, H) T is acyclic with $n - 1$ edges

Leaves of tree

- A vertex of degree 1 in a tree is called a leaf
- Theorem (1.14, H; Ex9, S1.3.2, H) Let T be a tree of order $n \geq 2$. Then T has at least two leaves
- (Ex3, S1.3.2, H) Let T be a tree with max degree Δ. Then T has at least Δ leaves
- (Ex10, S1.3.2, H) Let T be a tree of order $n \geq 2$. Then the number of leaves is

$$2 + \sum_{v:d(v)\geq 3} (d(v) - 2)$$

- (Ex8, S1.3.2, H) Every nonleaf in a tree is a cut vertex

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

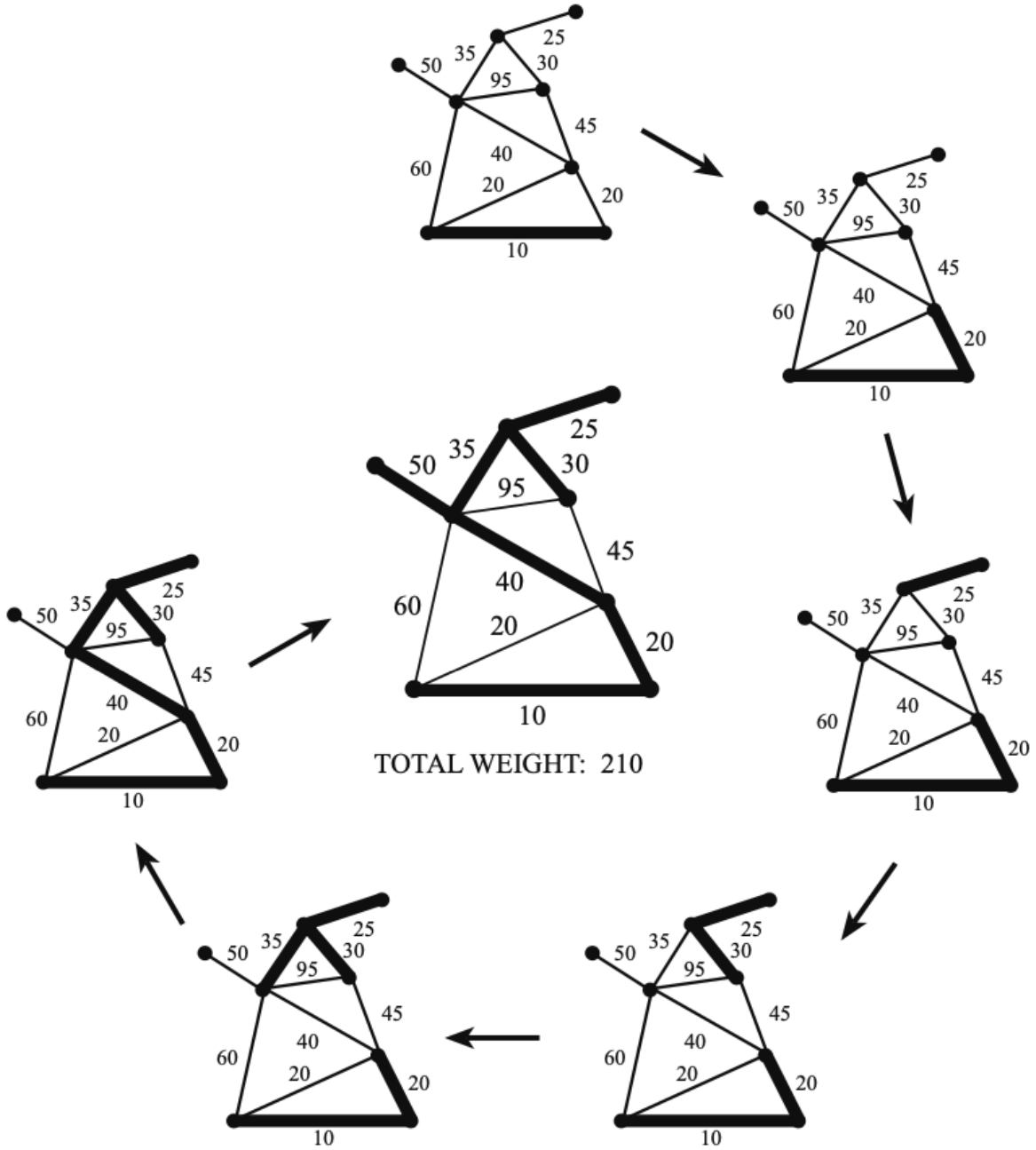


FIGURE 1.43. The stages of Kruskal's algorithm.

Theoretical guarantee of Kruskal's algorithm

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

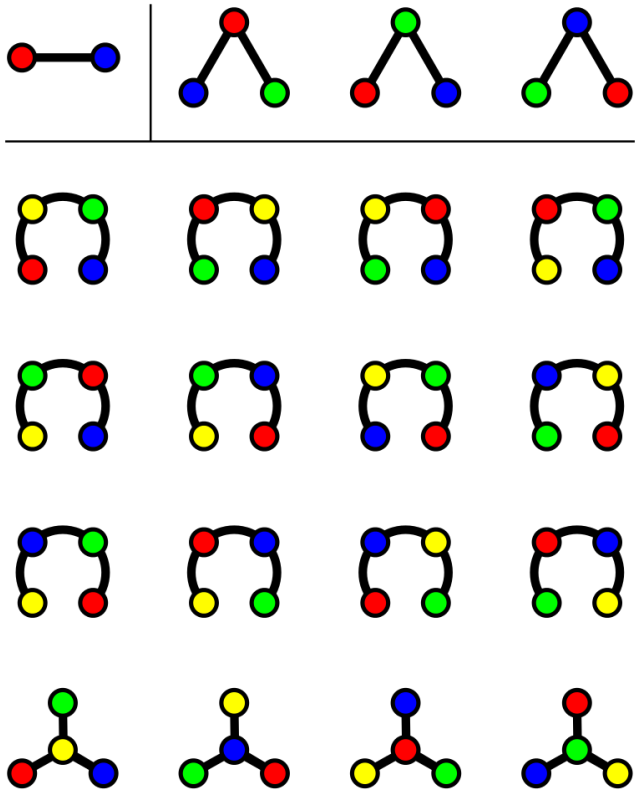
Prim's Algorithm

- 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



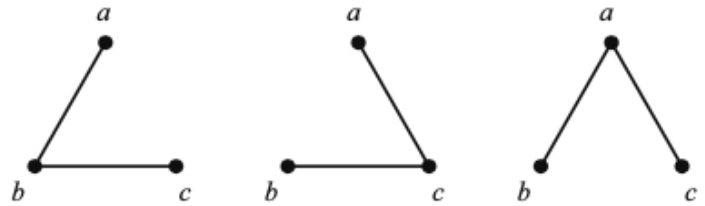


FIGURE 1.45. Labeled trees on three vertices.

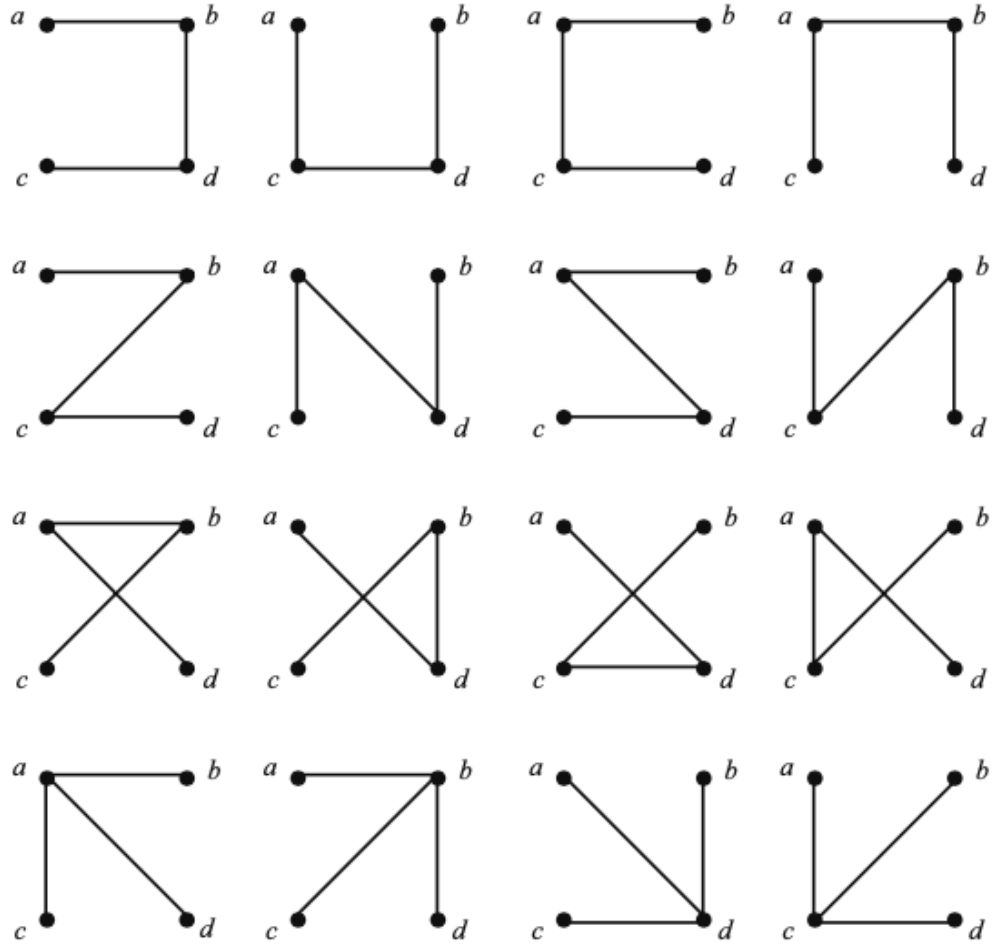


FIGURE 1.46. Labeled trees on four vertices.

Example

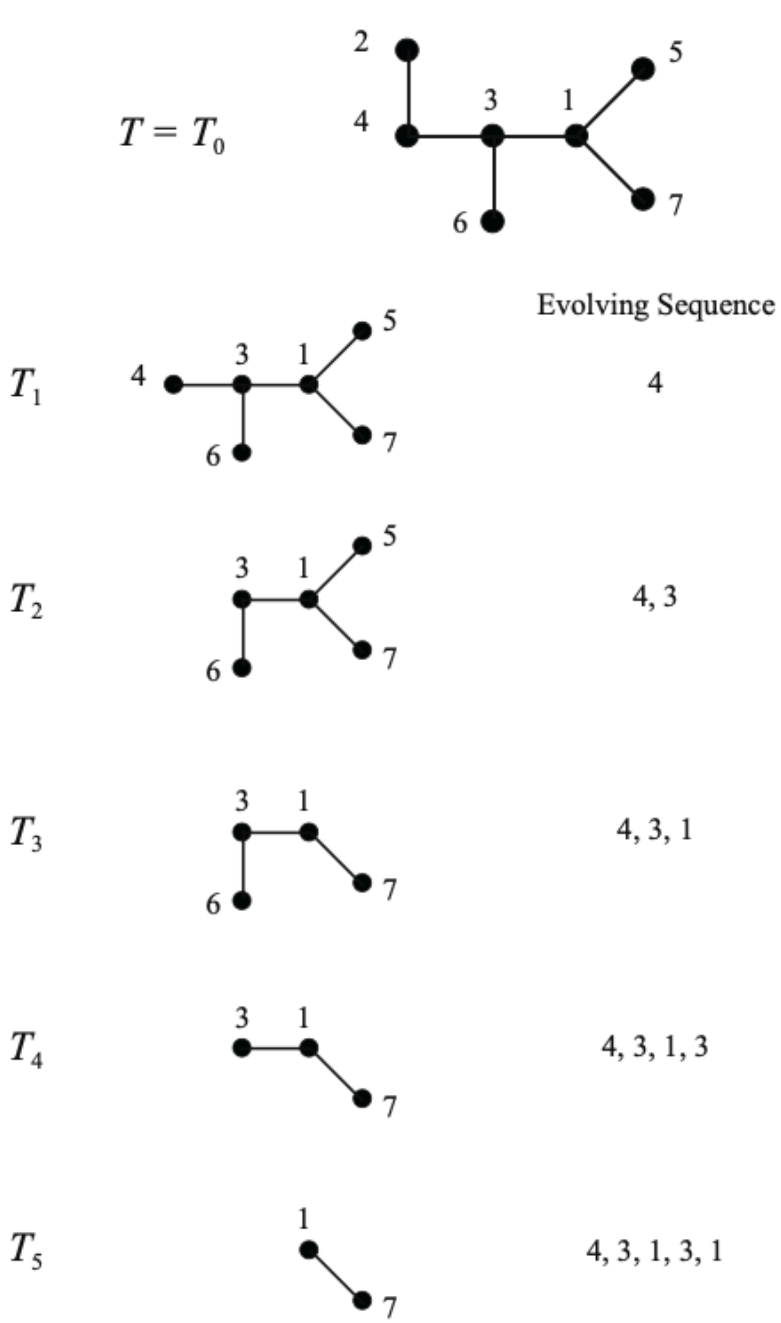


FIGURE 1.47. Creating a Prüfer sequence.

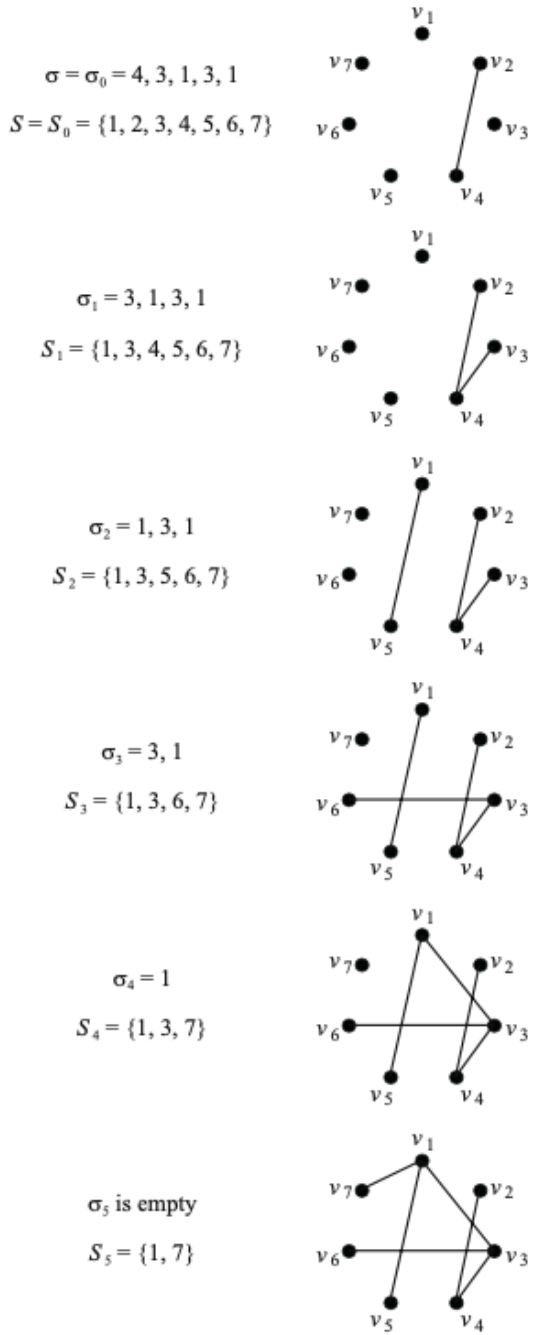
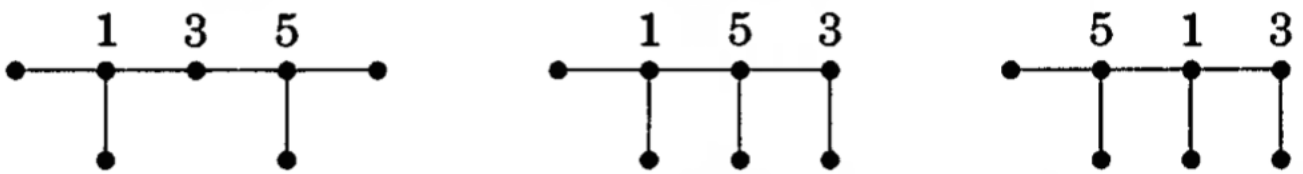


FIGURE 1.48. Building a labeled tree.

Trees with fixed degrees

- Corollary (2.2.4, W) Given positive integers $d_1, \dots, d_n$ 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

- Theorem (1.19, H) 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 $D - A$

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 (2.1.4, W) Among trees with n vertices, the Wiener index $D(T)$ is minimized by stars and maximized by paths, both uniquely

Huffman coding