Lecture 3: Trees

Shuai Li

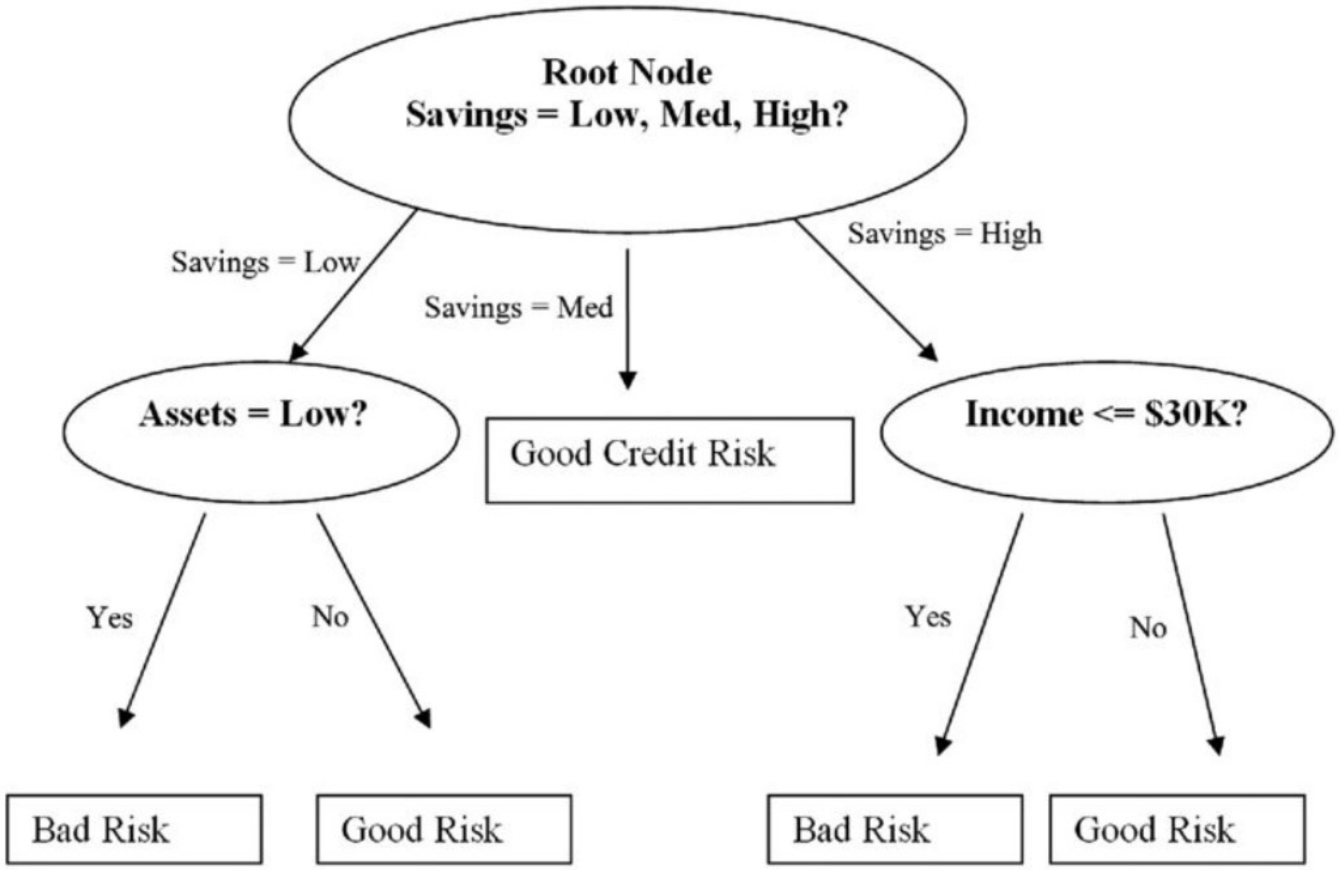
John Hopcroft Center, Shanghai Jiao Tong University

https://shuaili8.github.io

https://shuaili8.github.io/Teaching/CS3330/index.html

Trees

- A tree is a connected graph T with no cycles



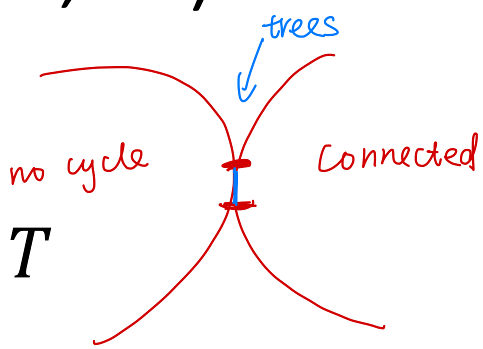
Properties

- Recall that Theorem (1.2.18, W, Kőnig 1936)
  A graph is bipartite $\Leftrightarrow$ it contains no odd cycle
- $\Rightarrow$(Ex 3, S1.3.1, H) A tree of order $n \geq 2$ is a bipartite graph
- Recall that Proposition (1.2.14, W)
  An edge e is a bridge $\Leftrightarrow$ e lies on no cycle of G
  - Or equivalently, an edge e is not a bridge $\Leftrightarrow$ e lies on a 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 connected, no cycle

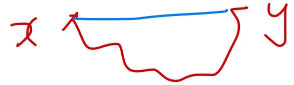
connected
no cycle
n-1 edges



  ⇔ Any two vertices of T are linked by a unique path in T

  ⇔ T is minimally connected
  - i.e. T is connected but $T - e$ is disconnected for every edge $e \in T$

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

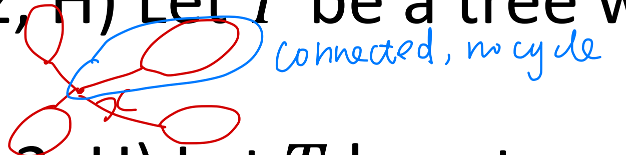


  ⇔ (Theorem 1.10, 1.12, H) T is connected with $n - 1$ edges

  ⇔ (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 *Take the longest path* $v_0\ v_1 \cdots v_k$ $d(v_0) = d(v_k) = 1$
- (Ex3, S1.3.2, H) Let T be a tree with max degree Δ. Then T has at least Δ leaves

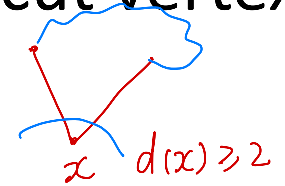


- (Ex10, S1.3.2, H) Let T be a tree of order $n \geq 2$. Then the number of leaves is

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

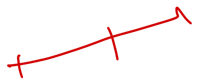
$$\sum_v d(v) = 2|E| = 2(n-1)$$
$$\Downarrow$$
$$n_1 + 2\cancel{n_2} + \sum_{v:d(v)\geq 3} d(v) = 2\Big(n_1 + \cancel{n_2} + \sum_{v:d(v)\geq 3} 1 \; - 1\Big)$$

- (Ex8, S1.3.2, H) Every nonleaf in a tree is a cut vertex
- Every leaf node is not a cut vertex



The center of a tree is a vertex or ‘an edge’

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



$T_0 := T$

$T_1 := T_0 - \text{leaves of } T_0$

$T_2 := T_1 - \text{leaves of } T_1$

$\vdots$

$T_r = K_1$ or K_2 (有限步停止)

$\text{center}(T) = \text{center}(T_r) = K_1$ or K_2

$\text{center}(T_i) = \text{center}(T_{i+1})$

① T, $n \geq 3$, leaf $v \notin \text{center}(T)$

unique neighbor w,

every path to v must go through w

$d(x, w) < d(x, v) \quad \forall x \neq v$

② $\forall v \in \text{non-leaf}(T) \quad \underset{x}{\text{argmax}}\, d_T(v, x) \subseteq \text{Leaves}(T)$

$$\max_{x \in T_i} d_{T_i}(x, v) = \max_{x \in T_{i+1}} d_{T_{i+1}}(x, v) + 1$$

$$\text{center}(T_i) = \underset{v \in \underbrace{T_i - \text{leaves}}_{T_{i+1}}}{\text{argmin}} \max_{x \in T_{i+1}} d_{T_i}(v, x) = \text{center}(T_{i+1})$$

Any tree can be embedded in a ‘dense’ graph

- 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

By induction. $k=0, k=1$ ✓

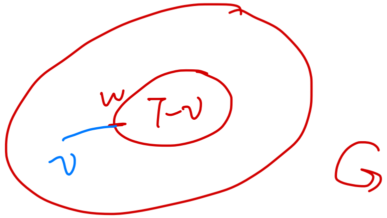
$k-1$ $(k \geq 2)$ T is a tree w/ at least 2 leaves. Take v as a leaf w/ $vw \in E$

$(T-v)$ is a tree w/ k vertices, $(k-1)$ edges

$f: T-v \hookrightarrow G$

$w \longmapsto f(w)$

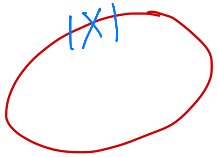
$d_G f(w) \geq k$



$\exists x \in N_G(f(w)) \setminus f(T-v)$, $f(v) =: w$

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

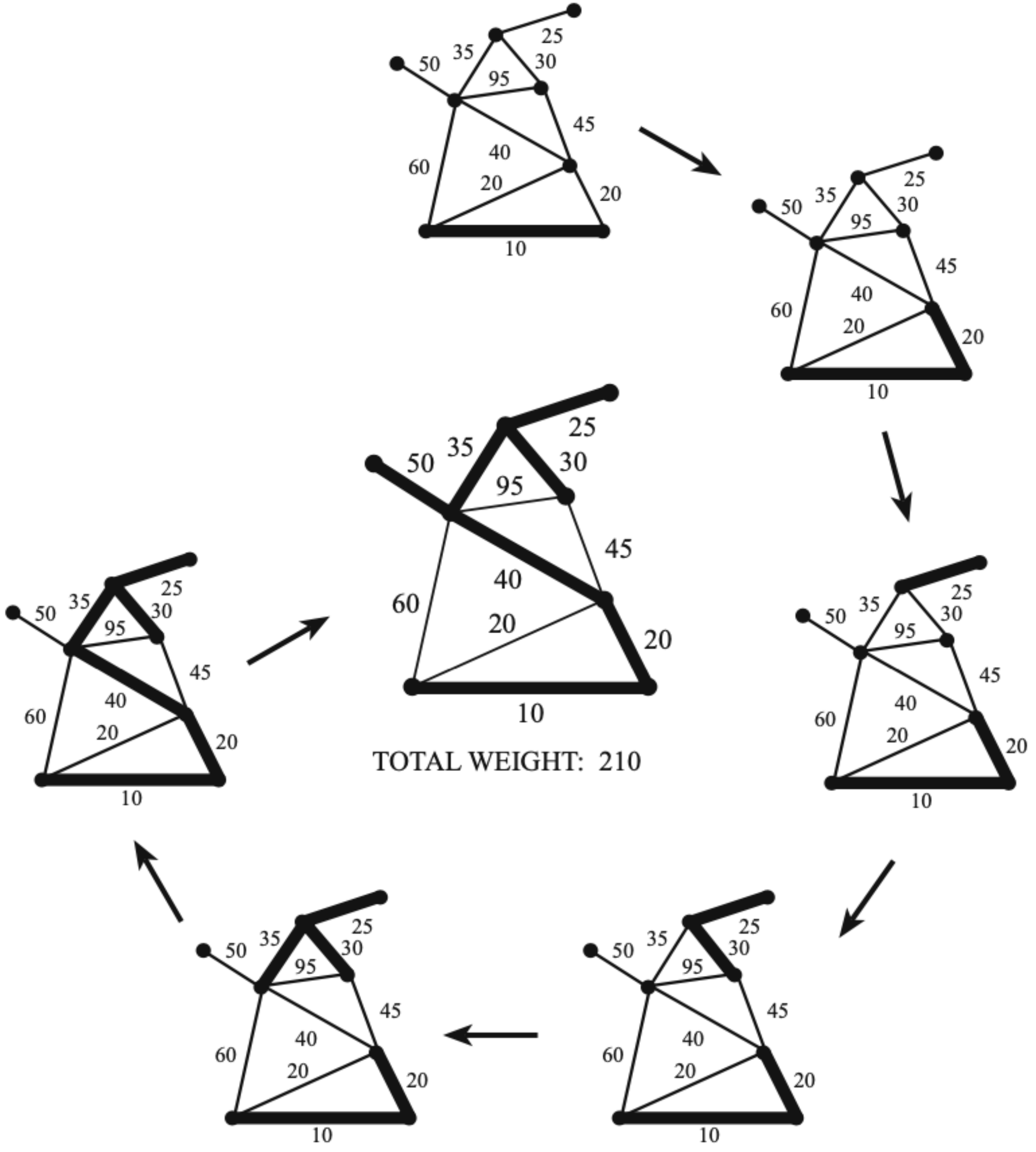


FIGURE 1.43. The stages of Kruskal's algorithm.

Theoretical guarantee of Kruskal's algorithm

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

G is connected $\Rightarrow$ the output is spanning tree. $e_1, \cdots, e_{n-1}$

反证法. T is not minimal.

Among all minimal spanning trees, T' is the one having the largest prefix edges

$e_1, \cdots, e_k$ ~~e_{k+1}~~

$T' + e_{k+1}$ contains a cycle C

$\exists e' \in C - T \Rightarrow T' + e_{k+1} - e' =: T''$ spanning tree

$w(e') \geq w(e_{k+1}) \Rightarrow w(T'') \leq w(T')$

T'' is minimal spanning tree but w/ more prefix $e_1, \cdots, e_k, e_{k+1}$. Contradiction!

Prim's Algorithm

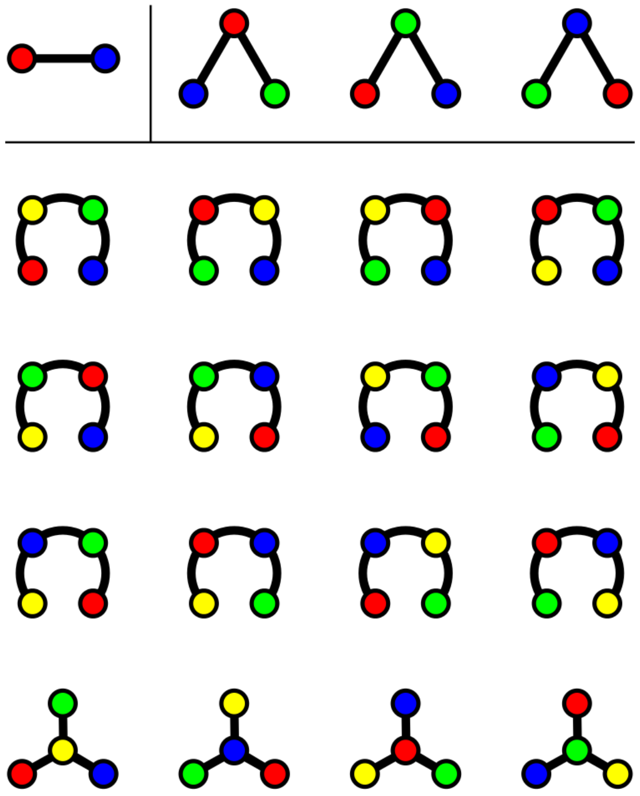
- Given: A connected, weighted graph G.

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

- Exercise (Ex2.3.10, W) Prim's algorithm produces a minimum-weight spanning tree of G

Cayley's tree formula

- Theorem (1.18, H; 2.2.3, W). There are n^{n-2} distinct labeled trees of order n



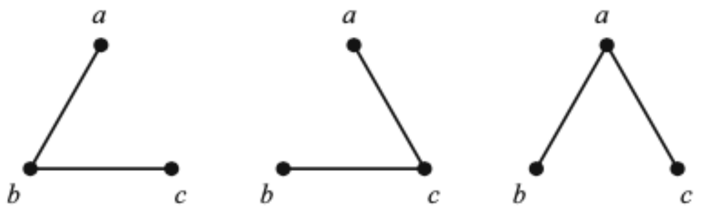


FIGURE 1.45. Labeled trees on three vertices.

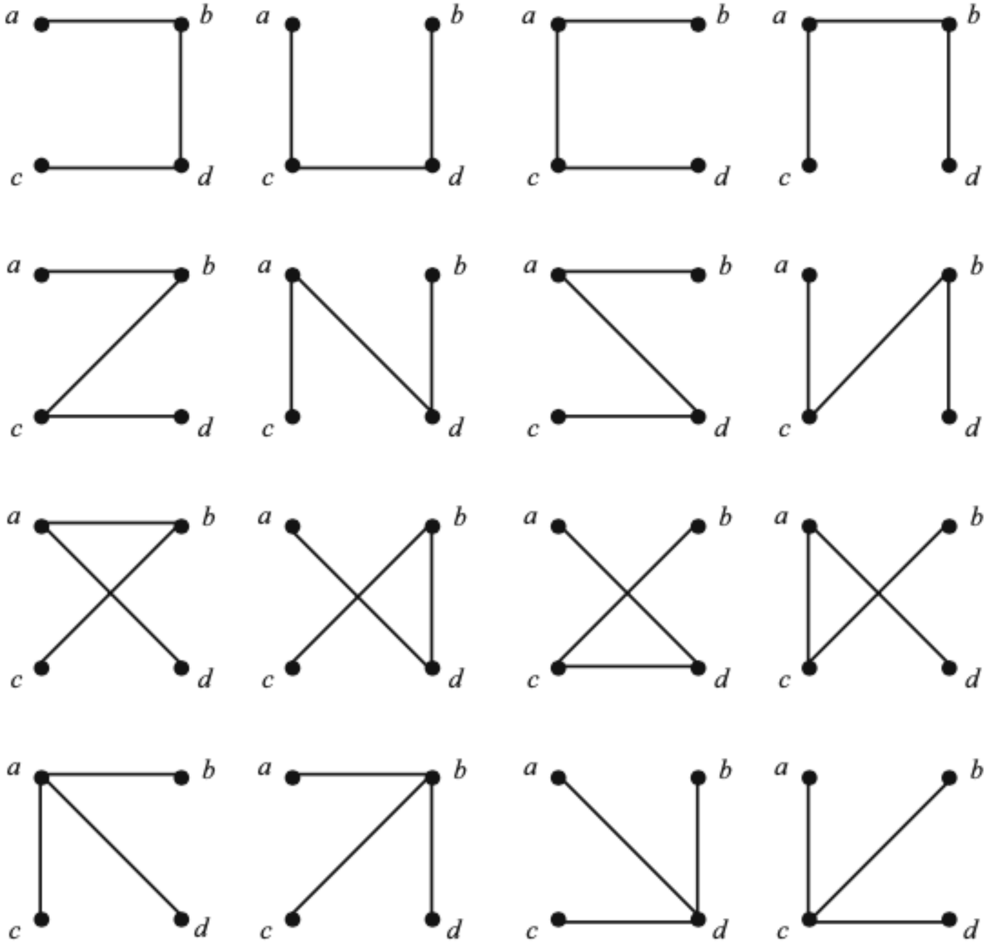


FIGURE 1.46. Labeled trees on four vertices.

Label K_n w/ $[n] := \{1, 2, \cdots, n\}$. How many spanning trees?

$P : \{\text{labeled tree of order } n\} \longrightarrow [n]^{n-2}$

$T \longmapsto P(T)$ Prüfer code

Given a spanning tree T

$T_1 := T$ $x_i \in T_i$ is the leaf w/ minimal index, $x_i y_i \in E$

$T_{i+1} = T_i - x_i y_i$

$$\begin{matrix} x_1 & x_2 & \cdots & x_{n-2} & x_{n-1} \\ y_1 & y_2 & & y_{n-2} & \underset{\sim}{y_{n-1}} = n \end{matrix}$$

① $y_{n-1} = n$

② Tree T_i : vertices $x_1 \; x_i \; x_{i+1} \; \cdots \; x_{n-2} \; x_{n-1}$

edges $y_1 \; y_i \; y_{i+1} \; \cdots \; y_{n-2} \; y_{n-1}$

$\forall v$ v appears exactly once in $x_1, \cdots, x_{n-2}, x_{n-1}, y_{n-1}$

$d(v)$ times in $x_1, \cdots, x_{n-2}, x_{n-1}, y_1, \cdots, y_{n-1}$

$\therefore$ v appears $d(v)-1$ times in $\underline{y_1 \cdots y_{n-2}} =: P(T)$

$x_1 := \min([n] - \{y_1, \cdots, y_{n-2}\})$

$$\begin{matrix} x_1 & & & & \\ y_1 & y_2 & \cdots & y_{n-2} & y_{n-1} = n \end{matrix}$$

$x_2 := \min([n] - \{x_1\} - \{y_2, \cdots, y_{n-2}\})$

$\vdots$

$x_i := \min([n] - \{x_1, \cdots, x_{i-1}\} - \{y_i, \cdots, y_{n-2}\})$

Example

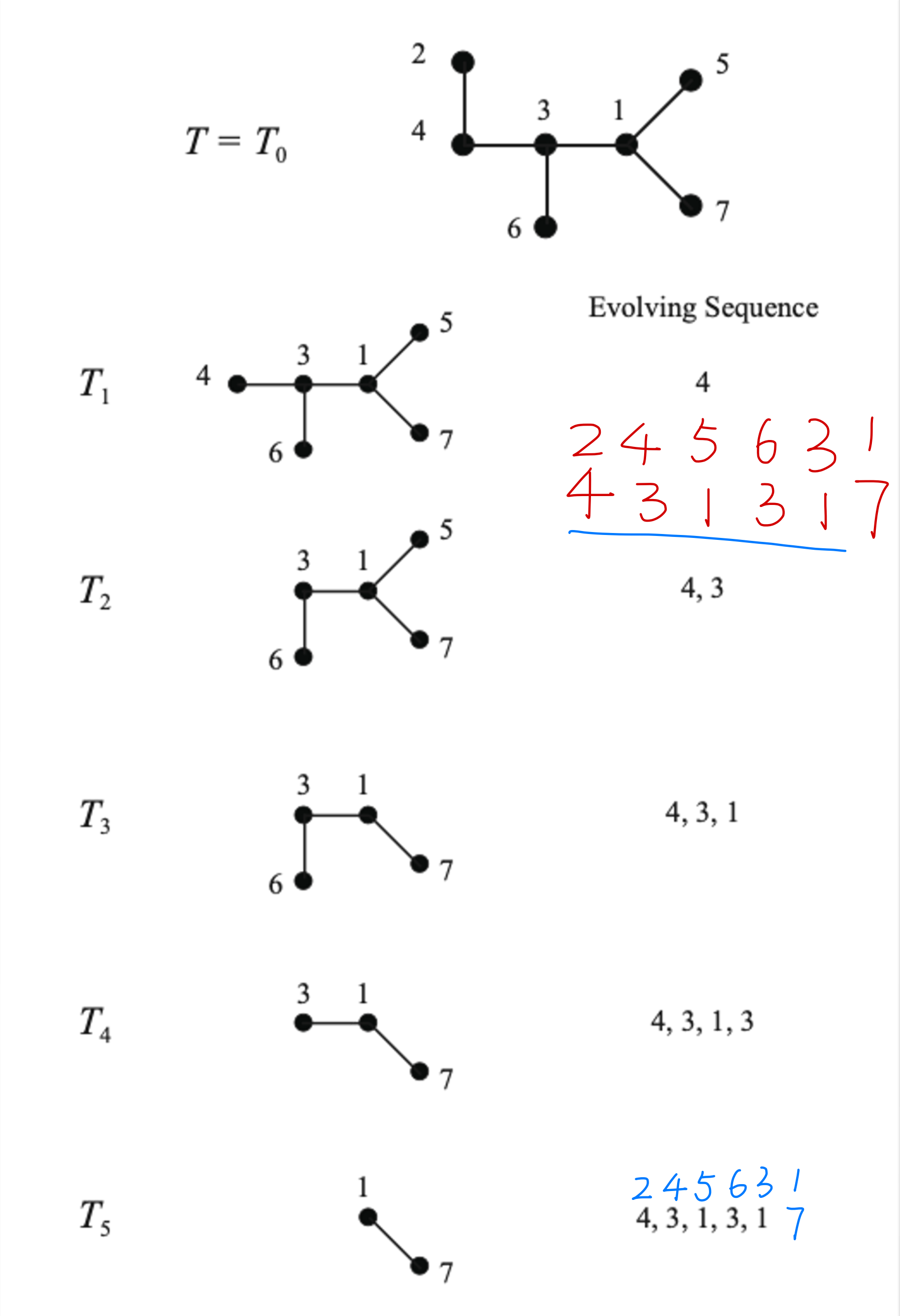


FIGURE 1.47. Creating a Prüfer sequence.

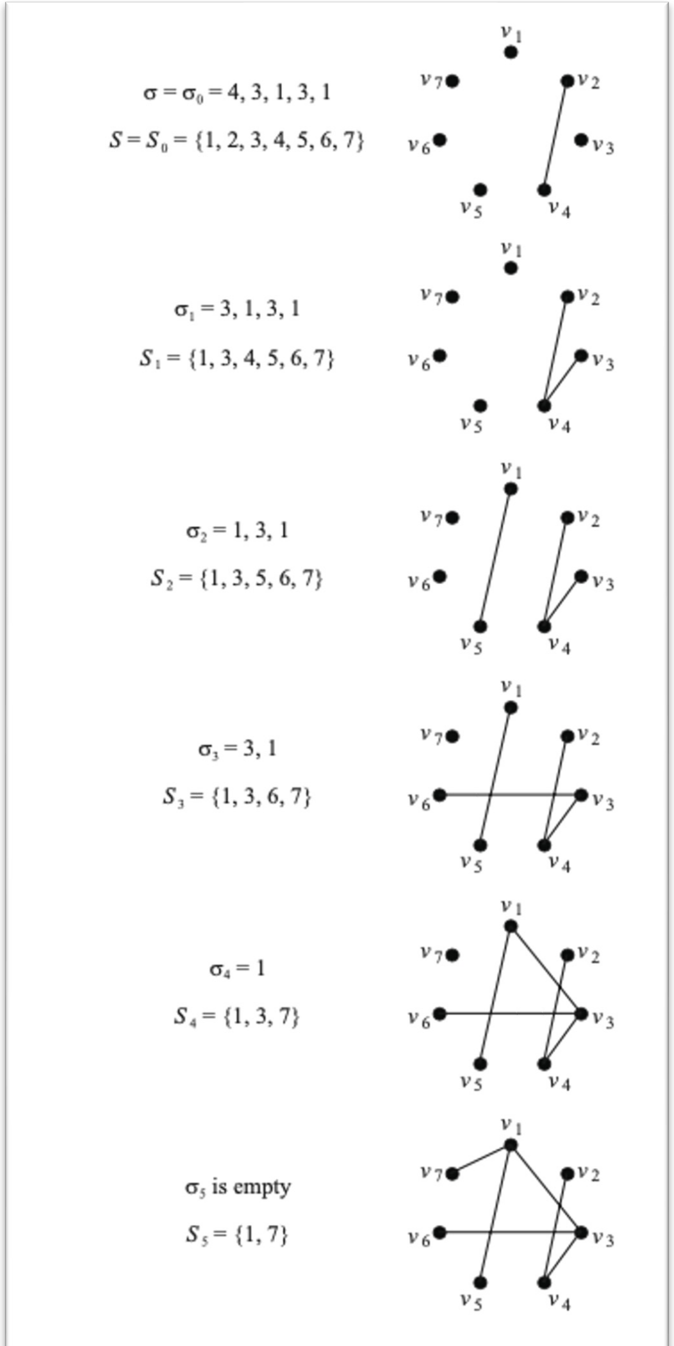
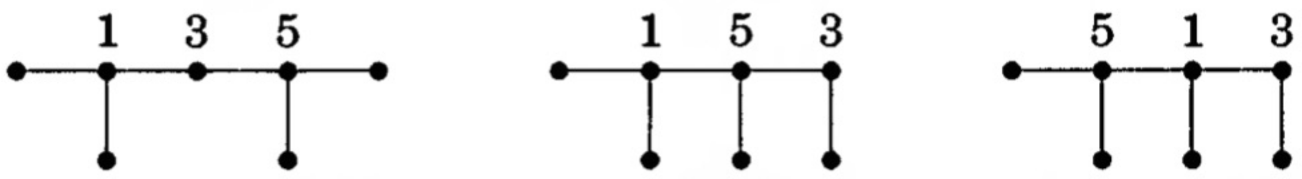


FIGURE 1.48. Building a labeled tree.

of trees with fixed degree sequence

- Corollary (2.2.4, W) Given positive integers $d_1, \overbrace{\ldots,}^{n-2} d_n$ 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 $(3,1,2,1,3,1,1)$



$\binom{4}{2} + \binom{4}{2}\times 2 + \binom{4}{2}\times 2 = ?$

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

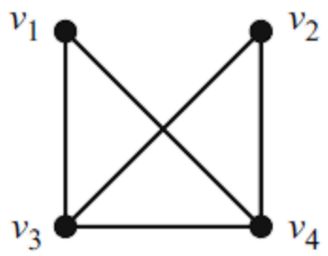
$$\mathbf{C} = \begin{pmatrix} +\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix}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

Matrix tree theorem

$$A = \begin{array}{cc} & j \\ i & \begin{pmatrix} & | & \\ - & 1 & - \\ & & 0 \end{pmatrix} \end{array} \qquad ij \in E(G) \qquad D = \begin{pmatrix} d(1) & & \\ & \ddots & \\ & & d(n) \end{pmatrix}$$

- 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 $D - A$ Laplacian matrix
- 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
- Exercise (Ex7, S1.3.4, H) Use the matrix tree theorem to prove Cayley's theorem

Example



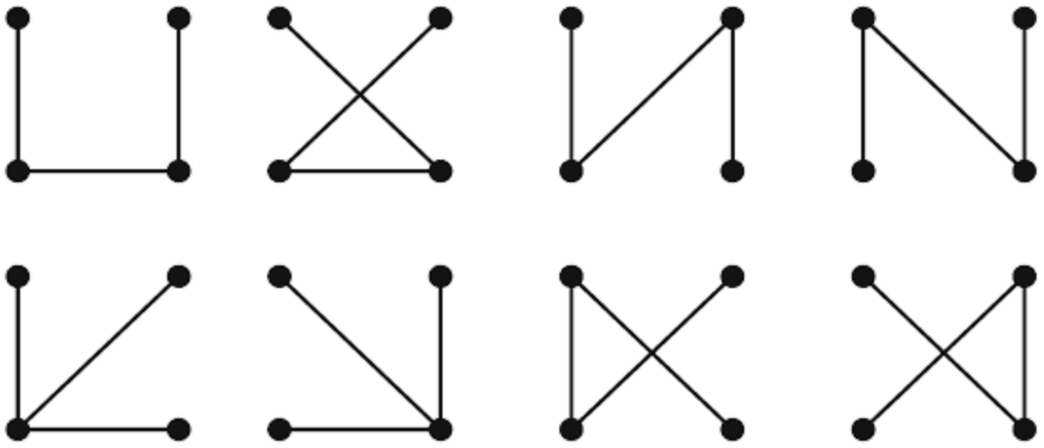


FIGURE 1.49. A labeled graph and its spanning trees.

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

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

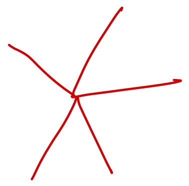
The $(1,1)$ cofactor of $D - A$ is

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

Score one for Kirchhoff!

- Exercise (Ex6, S1.3.4, H) Let e be an edge of K_n. Use Cayley's Theorem to prove that $K_n - e$ has $(n-2)n^{n-3}$ spanning trees

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
- Over all connected n-vertex graphs, $D(G)$ is minimized by K_n and maximized (2.1.16, W) by paths
  - (Lemma 2.1.15, W) If H is a subgraph of G, then $d_G(u,v) \leq d_H(u,v)$

· Stars Every tree has $(n-1)$ edges.

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

To show uniqueness, consider a leaf $x \in T$ w/ neighbor v

all other vertices must have distance 2 from x (by minimality)

$\therefore$ all other vertices must be neighbors of v $\therefore T$ is a star

· Path $D(P_{n-1})$

$n=1$ P_0 · $D(P_0) = 0$

$n=2$ P_1  $D(P_1) = 1 = \binom{3}{3}$

$n=3$ P_2  $D(P_2) = 1+2+1 = 4 = \binom{4}{3}$

$$D(P_n) = 1+2+\cdots+n + D(P_{n-1})$$

$$= \binom{n+1}{2} + \binom{(n-1)+2}{3} = \binom{n+2}{3}$$

By induction $n = 1, 2, 3$ ✓

$n-1$ $(n \ge 4)$ Let u be a leaf

$$D(T) = \underbrace{D(T-u)}_{\le D(P_{n-2})} + \underbrace{\sum_{v \in T} d(u,v)}$$

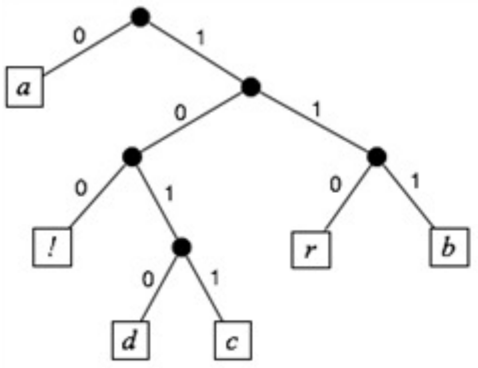
For any shortest path from u to a farthest vertex (length k), there is a list of sequences to u $(1, 2, \cdots, k)$.

Any repetition would make Σ smaller

Uniqueness: If T is not a path, $\exists$ repetition

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 (expected) total length
- Prefix-free coding: no code word is an initial portion of another



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

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

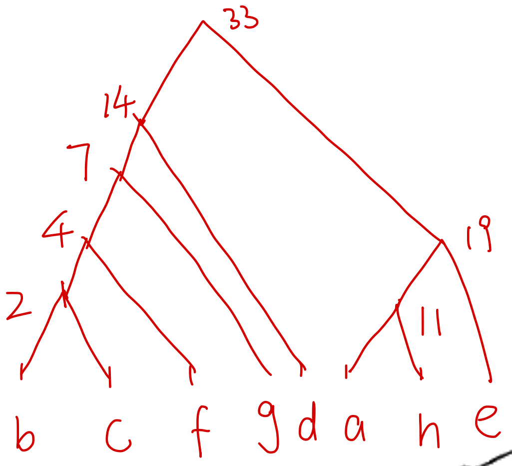
- Example: 11001111011

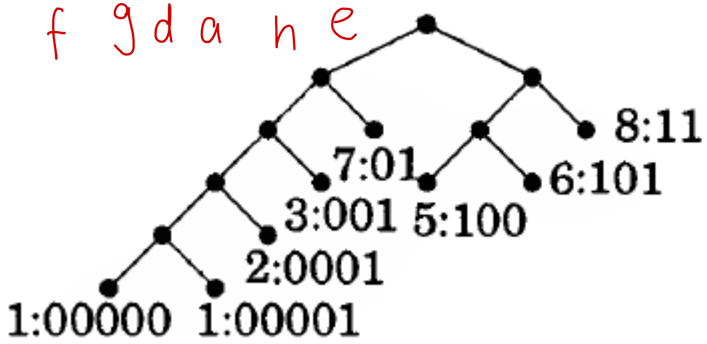
r a b c

Huffman's Algorithm (2.3.13, W)

- 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

By induction. $n=2$ $\quad 0, 1 \quad L=1$

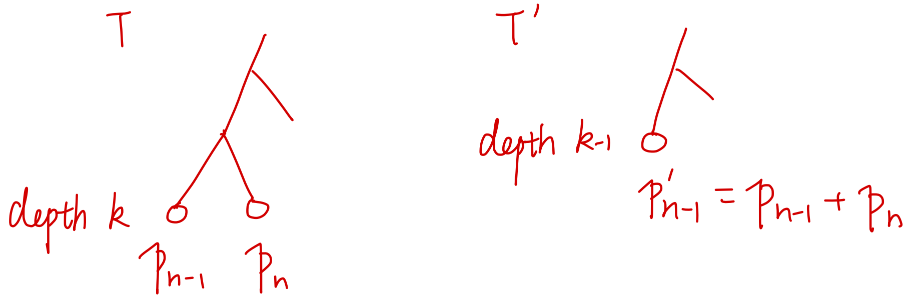
$n-1$ $(n \geq 3)$ ✓

Each prefix-free code is represented by a binary tree

Suppose T is an optimal tree for $p_1 \geq p_2 \geq \cdots \geq p_n$

∘ WLOG. p_{n-1} and p_n are siblings at greatest depth

∘ $T' = T -$ leaves of p_{n-1}, p_n



$L(T') = L(T) - k \cdot (p_{n-1} + p_n) + (k-1)\, p'_{n-1}$

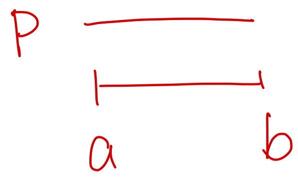
$= L(T) - (p_{n-1} + p_n)$

$\therefore T'$ is optimal. By induction hypothesis, $L(T') = L(HC_{n-1})$

Also since the first step of HC is to merge p_{n-1} and p_n to p'_{n-1}

$L(T) = L(T') \cdots = L(HC_n)$

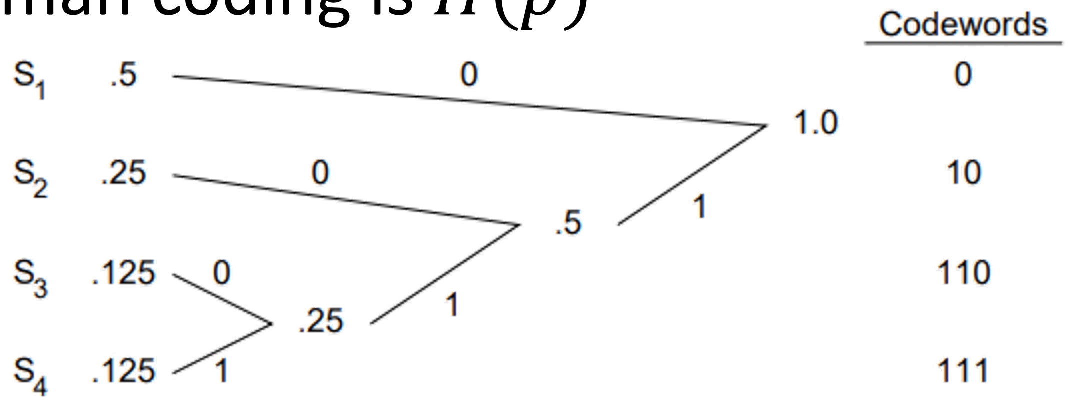
Huffman coding and entropy



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

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

- Exercise (Ex2.3.31, W) $H(p) \le$ average length of Huffman coding $\le H(p) + 1$
- Exercise (Ex2.3.30, W) When each p_i is a power of ½, average length of Huffman coding is $H(p)$



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

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

Summary

Shuai Li

https://shuaili8.github.io

- Trees
  - Is bipartite, edges are bridge; Equivalent definitions
  - Leaves, # of leaves, cut vertex; Center is a vertex or `an edge'
  - Any tree can be embedded in a 'dense' graph
- Spanning Tree
  - Every connected graph has a spanning tree
  - Minimal spanning tree, Kruskal's Algorithm (with guarantee), Prim's algorithm
  - Cayley's tree formula, Prüfer code, # of trees with fixed degree sequence
  - Matrix tree theorem
- Wiener index
  - Among trees, minimized by starts, maximized by paths
  - Among connected graphs, minimized by complete graphs, maximized by paths
- Hoffman coding
  - Algorithm, optimality, entropy

Questions?