Lecture 2, Trees

Qi Chen, Jingliang Gao

Fall , 2023

Trees with label

Theorem (Cayley)

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

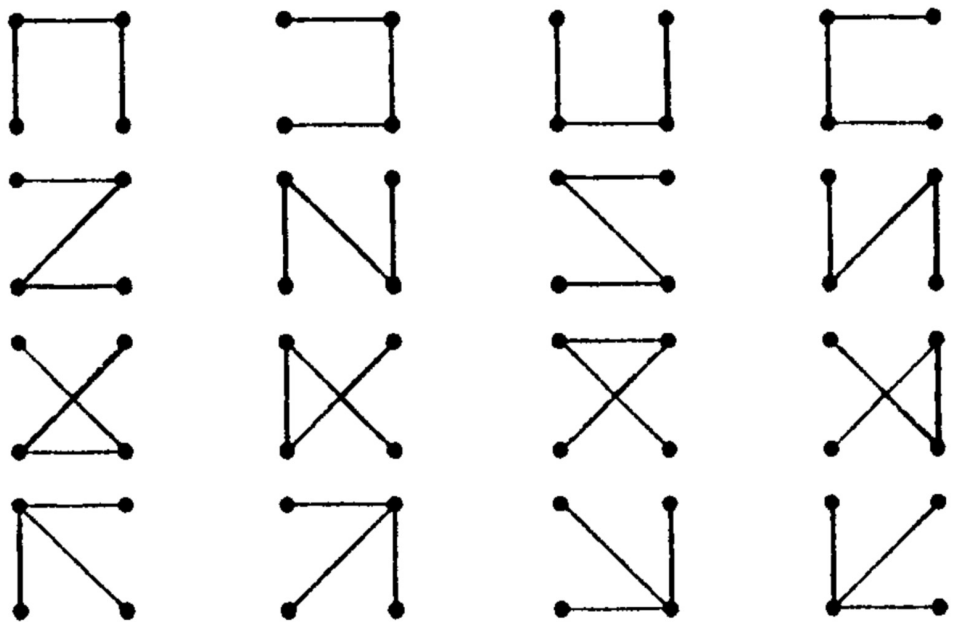
Trees with label

Theorem (Cayley)

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

Example

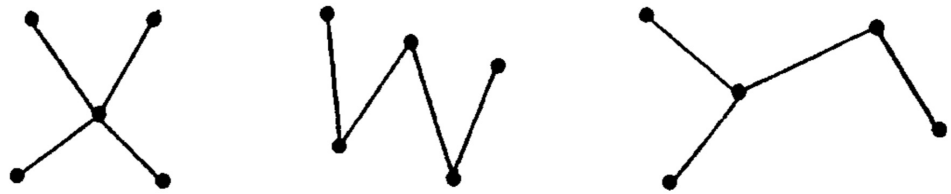
Here are the 16 labeled trees on four vertices:



Nonisomorphic trees

Example

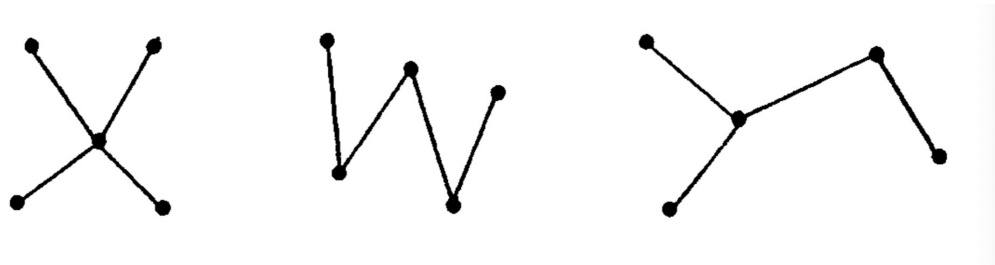
There are the three nonisomorphic trees on five vertices:



Nonisomorphic trees

Example

There are the three nonisomorphic trees on five vertices:



Proposition

The number of labeled trees on n vertices isomorphic to a specific tree T is $\frac{n!}{|\mathrm{Aut}(T)|}$, where $\mathrm{Aut}(T)$ is the automorphism group of T.

Forest, spanning tree and weighted graph

Definition

A graph with no circles as subgraphs is called a forest. Each component $C_1, C_2, \cdots, C_k$ of a forest G is a tree, so if a forest with n vertices has k components, it has $n - k$ edges

Forest, spanning tree and weighted graph

Definition

A graph with no circles as subgraphs is called a forest. Each component $C_1, C_2, \cdots, C_k$ of a forest G is a tree, so if a forest with n vertices has k components, it has $n - k$ edges

Definition

A spanning tree of a connected graph G is a spanning subgraph of G that is a tree.

Forest, spanning tree and weighted graph

Definition

A graph with no circles as subgraphs is called a forest. Each component $C_1, C_2, \cdots, C_k$ of a forest G is a tree, so if a forest with n vertices has k components, it has $n - k$ edges

Definition

A spanning tree of a connected graph G is a spanning subgraph of G that is a tree.

Definition

A weighted graph is a graph G together with a function associating a real number $c(e)$ (usually nonnegative) to each edge e, called its length or cost according to context.

Cost of a spanning tree

$$c(T) = \sum_{e \in T} c(e).$$

Greedy algorithm for searching a spanning tree with minimum cost

Definition

A set S of edges of a graph G is called independent when the spanning subgraph with edge set S (denoted $G : S$) is a forest.

Greedy algorithm for searching a spanning tree with minimum cost

Definition

A set S of edges of a graph G is called independent when the spanning subgraph with edge set S (denoted $G : S$) is a forest.

Greedy algorithm(Kruskal)

Let G be a connected weighted graph with n vertices.

1. For $i = 0$, let S_0 be the empty graph with n vertices
2. For each i, let $S_i = \{e_1, e_2, \cdots, e_i\}$ be independent edges so that $G : S_i$ has $n - i$ component.
3. If $i < n - 1$, let e_{i+1} be an edge with ends in different components of G of $G : S_i$ and whose cost is minimum.
4. Stop when we have chosen $n - 1$ edges.

Theorem 2.2

Theorem

With $e_1, \cdots, e_{n-1}$ chosen as above, the spanning tree $T_0 := G : \{e_1, \cdots, e_{n-1}\}$ has the property that $c(T_0) \leq c(T)$ for any spanning tree T.

Theorem 2.2

Proof.

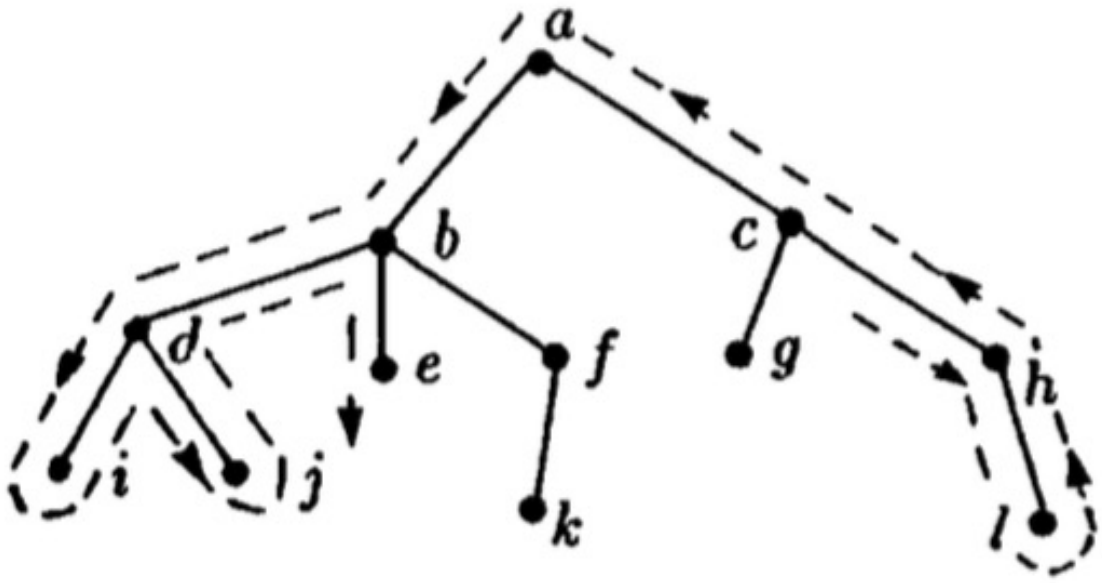
- Let $\{a_1, a_2, \cdots, a_{n-1}\}$ be the edge set of a tree T, numbered so that $c(a_1) \leq c(a_2) \leq \cdots, \leq c(a_{n-1})$.
- We claim that $c(e_i) \leq c(a_i)$ for each $i = 1, 2, \cdots, n-1$.
- If this is false, then

$$c(e_k) > c(a_k) \geq c(a_{k-1}) \geq \cdots \geq c(a_1)$$

  for some k.
- Each $a_i, i = 1, 2, \cdots, k$ should has both ends in the same components of $G : S_{k-1}$.
- the number of components of $G : \{a_1, a_2, \cdots, a_k\}$ is at least the number $n - k + 1$ of the components of $G : S_{k-1}$
- It contradicts the fact that $G : \{a_1, a_2, \cdots, a_k\}$ is independent.

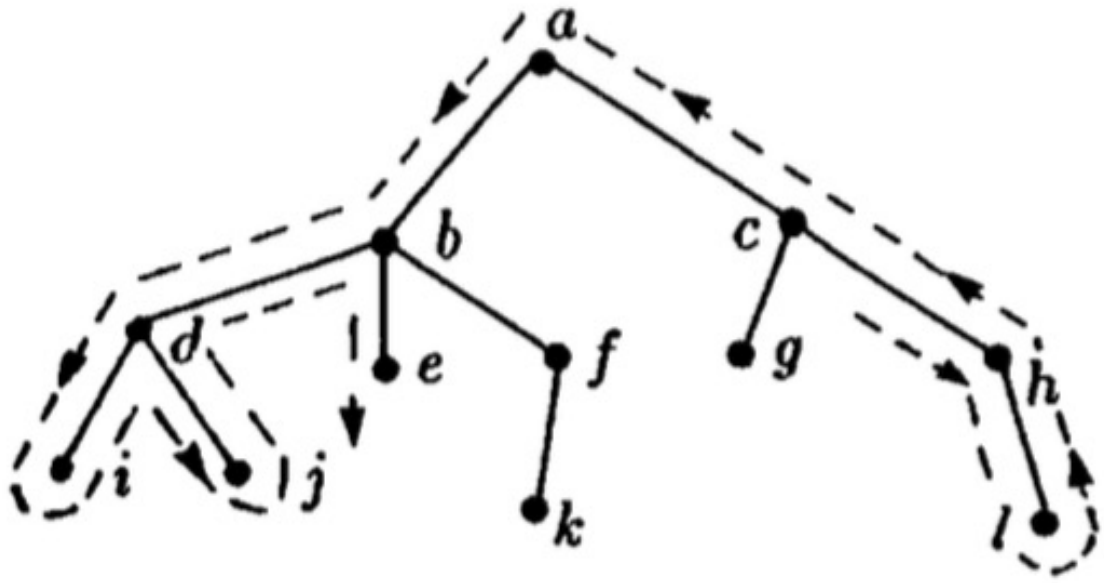
Depth-first search and breadth-first search in a rooted tree

A tree with a distinguished vertex-the root-is called a rooted tree.



Depth-first search and breadth-first search in a rooted tree

A tree with a distinguished vertex-the root-is called a rooted tree.



- In a depth-first search, one search along the walk *abdidjdbebfk . . . lhca*
- in a breadth-first search, one search alphabetically in the above figure.

Depth-first search tree of a finite connected graph G

Construction of the depth-first search tree T of G

1. Pick a virtex v_0 and start with the tree T_0 with vertex v_0 and no edges.
2. Proceed inductively: once vertices $v_0, v_1, v_2, \cdots, v_k$ and a tree T_k with exactly those vertices and some of the edges of G have been chosen, let l be the largest index $\leq k$ so that v_l is adjacent to some vertex not in T_k.
3. Call that new vertex v_{k+1} and add it and the edge $\{v_l, v_{k+1}\}$ to T_k to obtain a tree T_{k+1}.
4. Stop when T_k is a spanning tree of G and let $T = T_k$.

Ancestors and descendants in a rooted tree

Definition

Given a vertex x of a rooted tree with root v_0, the ancestors of x are the vertices traversed by the (unique) path from x to the root v_0. The first vertex other than x on that path is the parent of x. If x is an ancestor of y, we also say that y is a descendant of x.

Proposition 2.3

Proposition

If vertices x and y are adjacent in G, then one of them is a descendant of the other in any depth-first search tree T of G.

Proof.

- Asumme $x = v_k$ and $y = v_l$ with $k \leq l$.
- If $l = k + 1$, done.
- Otherwise, inductively, for all $v_{k'}$ with $k < k' < l$ must be descendant(s) of x.
- Then T_l is obtained by adding $\{v_{k'}, y\}$ to T_{l-1}, where $k \leq k' < l$.

Isthmus

Definition

An isthmus (or a bridge) of a connected graph G is an edge whose deletion results in a disconnected graph.

Remark

Note that every edge in a tree is an isthmus.

Proposition 2.4

Proposition

Let $\{x, y\}$ be an edge of T which is not an isthmus in G; say x is the parent of y. Then there is an edge in G, but not in T joining some descendant a of y and some ancestor b of x.

Proof.

- Let D be the set of descendants of y which including y itself, so $x \notin D$ and $y \in D$.
- As $\{x, y\}$ is not an isthmus, there exist another edge $\{a, b\}$ of G with $a \in D$ and $b \notin D$, and $\{a, b\} \notin T$
- According to Prop 2.3, a is a descendant of b.
- The unique path from a to v_0 must pass through y, x and b, which implies that b is an ancestor of x.
- Then $\{a, b\}$ is what we want.

Strongly connectedness

Definition

- Any directed graph obtained from an undirected graph G by assigning a direction to each edge of G is called an orientation of G.
- A walk in a digraph D may be called strong when each edge is traversed by the walk according to its direction, i.e. from its tail to its head.
- The digraph D is strongly connected when for any two vertices x, y, there is a strong walk from x to y.

Theorem 2.5

Theorem

Let G be a finite connected graph without isthmuses. Then G admits a strong orientation, i.e. an orientation that is a strongly connected digraph D.

Construction of D from G.

- Find a depth-first search tree T of G and numbering $\{v_1, v_2, \cdots, v_n\}$ of the vertices of G.
- If $\{v_i, v_j\}$ is in T, direct it from v_i to v_j, i.e., (v_i, v_j) is an edge of D.
- If $\{v_i, v_j\}$ is not in T, direct it from v_j to v_i, i.e., (v_j, v_i) is an edge of D.

Proof of Theorem 2.5

Proof of the strongly connectedness of D .

- There is a strong walk from v_0 to any vertex x of D, say v_k.
- For any v_k, by Prop. 2.4, there exists an edge (a, b) with descendant a of v_k and ancestor b of v_k. Let $b = v_i$, then $i < k$.
- Appending the strong walk from v_k to a and (a, v_i), we obtain a strong walk from v_k to v_i.
- If $i = 0$, done. Otherwise, repeat until reach v_0.

Thank you!