Trees and distance

Lecture 9

Characterization of trees

Theorem 2.2 (A characterization of trees): Let $n \geq 1$. For an n-vertex graph G, the following are equivalent

(A) G is connected and has no cycles.

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

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

(D) For any $u, v \in V(G)$, G has exactly one u, v-path.

(F) Adding to G any edge creates a graph with exactly one cycle.

Distances in graphs

Let G be a graph and let $u, v \in V(G)$.
If u and v are in the same component of G, then the **distance** from u to v is the length of the shortest u, v-path in G, and we write $d_G(u, v)$ for this (or often just $d(u, v)$). If u and v are in different components, then we define $d_G(u, v) = \infty$.

Distances in graphs

Let G be a graph and let $u, v \in V(G)$.
If u and v are in the same component of G, then the **distance** from u to v is the length of the shortest u, v-path in G, and we write $d_G(u, v)$ for this (or often just $d(u, v)$). If u and v are in different components, then we define $d_G(u, v) = \infty$.

Note that distance in undirected graphs satisfies the axioms of a metrics: it is symmetric, satisfies triangle inequality, and $d(u, u) = 0$.

Distances in graphs

Let G be a graph and let $u, v \in V(G)$.
If u and v are in the same component of G, then the **distance** from u to v is the length of the shortest u, v-path in G, and we write $d_G(u, v)$ for this (or often just $d(u, v)$). If u and v are in different components, then we define $d_G(u, v) = \infty$.

Note that distance in undirected graphs satisfies the axioms of a metrics: it is symmetric, satisfies triangle inequality, and $d(u, u) = 0$.

The **eccentricity** of u in G, denoted $\mathrm{ecc}(u)$ or $\epsilon(u)$ is the maximum distance from u to another vertex in G, or

$$\mathrm{ecc}(u) = \max_{v \in V(G)} d(u, v).$$

The **diameter** of G, $\mathrm{diam}(G)$, is defined as

$$\mathrm{diam}(G) = \max_{u \in V(G)} \mathrm{ecc}(u) = \max_{u,v \in V(G)} d(u, v).$$

The **diameter** of G, $\operatorname{diam}(G)$, is defined as

$$\operatorname{diam}(G) = \max_{u \in V(G)} \operatorname{ecc}(u) = \max_{u,v \in V(G)} d(u, v).$$

Finally, the radius, $\operatorname{rad}(G)$ is defined to be

$$\operatorname{rad}(G) = \min_{u \in V(G)} \operatorname{ecc}(u).$$

It could be that the radius and diameter of a graph are the same.

The **diameter** of G, $\mathrm{diam}(G)$, is defined as

$$\mathrm{diam}(G) = \max_{u \in V(G)} \mathrm{ecc}(u) = \max_{u,v \in V(G)} d(u, v).$$

Finally, the radius, $\mathrm{rad}(G)$ is defined to be

$$\mathrm{rad}(G) = \min_{u \in V(G)} \mathrm{ecc}(u).$$

It could be that the radius and diameter of a graph are the same.

The **center** of a graph G is the induced subgraph of G whose vertex set is the set of all vertices of eccentricity $\mathrm{rad}(G)$.

The center could be the whole graph.

Theorem 2.3 (Jordan, 1869): The center of any tree is either a vertex or two adjacent vertices.

Theorem 2.3 (Jordan, 1869): The center of any tree is either a vertex or two adjacent vertices.

Proof. **Base:** $n \leq 2$.
Induction step. Suppose the theorem holds for all trees with less than n vertices. Take any tree T with n vertices.

Let L be the set of leaves in T and $T' = T - L$.

Theorem 2.3 (Jordan, 1869): The center of any tree is either a vertex or two adjacent vertices.

Proof. **Base:** $n \leq 2$.
Induction step. Suppose the theorem holds for all trees with less than n vertices. Take any tree T with n vertices.

Let L be the set of leaves in T and $T' = T - L$.

By Lemma 2.1 (b), T' is a tree. We claim that

$$\text{for each } u \in V(T'),\ \epsilon_{T'}(u) = \epsilon_T(u) - 1. \tag{1}$$

Indeed, each longest path in T starting from u ends at a leaf (which is not in T'). This shows inequality $\leq$.
On the other hand, if $uv_1v_2 \ldots v_{k-1}v_k$ is a longest path in T starting from u, then all vertices $u, v_1, v_2 \ldots, v_{k-1}$ are in T'. Hence we also have $\geq$.

By (1), the center of T' is the same as in T.

This proves Theorem 2.3.

Coding of labeled trees

Among ways to code a graph are adjacency and incidency matrices. For labeled trees, there are nicer and shorter ways to code.

Consider the following procedure for a tree T with vertex set $\{1, \ldots, n\}$:

Prüfer algorithm. Let $T_0 = T$. For $i = 1, \ldots, n-1$,
(a) let b_i be the smallest leaf in T_{i-1},
(b) denote by a_i the neighbor of b_i in T_{i-1}, and
(c) let $T_i = T_{i-1} - b_i$.

Coding of labeled trees

Among ways to code a graph are adjacency and incidency matrices. For labeled trees, there are nicer and shorter ways to code.

Consider the following procedure for a tree T with vertex set $\{1, \ldots, n\}$:

Prüfer algorithm. Let $T_0 = T$. For $i = 1, \ldots, n-1$,
(a) let b_i be the smallest leaf in T_{i-1},
(b) denote by a_i the neighbor of b_i in T_{i-1}, and
(c) let $T_i = T_{i-1} - b_i$.

The Prüfer code of T is the vector $(a_1, \ldots, a_{n-2})$.

Properties of Prüfer algorithm

(P1) $a_{n-1} = n$.

Properties of Prüfer algorithm

(P1) $a_{n-1} = n$.

(P2) Any vertex of degree s in T appears in $(a_1, \ldots, a_{n-2})$ exactly $s - 1$ times.

Properties of Prüfer algorithm

(P1) $a_{n-1} = n$.

(P2) Any vertex of degree s in T appears in $(a_1, \ldots, a_{n-2})$ exactly $s - 1$ times.

(P3) $b_i = \min\{k : k \notin \{b_1, \ldots, b_{i-1}\} \cup \{a_i, a_{i+1}, \ldots, a_{n-2}\}\}$ for each i.

Properties of Prüfer algorithm

(P1) $a_{n-1} = n$.

(P2) Any vertex of degree s in T appears in $(a_1, \ldots, a_{n-2})$ exactly $s - 1$ times.

(P3) $b_i = \min\{k : k \notin \{b_1, \ldots, b_{i-1}\} \cup \{a_i, a_{i+1}, \ldots, a_{n-2}\}\}$ for each i.

Proofs. (P1) follows from the fact that we always have a leaf distinct from n.

(P2) follows from the facts that at the moment some k appears in $(a_1, \ldots, a_{n-2})$, its degree decreases by 1 and for $s \geq 3$ the neighbors of leaves in s-vertex trees are not leaves.

(P3) follows from the algorithm and (P2).

Theorem 2.4 (Prüfer, 1918): Every vector $(a_1, \ldots, a_{n-2})$ with $a_i \in \{1, \ldots, n\}$ for each i is the Prüfer code of exactly one labeled n-vertex tree.

Theorem 2.4 (Prüfer, 1918): Every vector $(a_1, \ldots, a_{n-2})$ with $a_i \in \{1, \ldots, n\}$ for each i is the Prüfer code of exactly one labeled n-vertex tree.

Proof. **Uniqueness**. By (P1) we know $a_{n-1} = n$. Then by (P3), we can reconstruct b_i for all $1 \leq i \leq n-1$. Thus the edges are $a_1 b_1, \ldots, a_{n-1} b_{n-1}$.

Theorem 2.4 (Prüfer, 1918): Every vector $(a_1, \dots, a_{n-2})$ with $a_i \in \{1, \dots, n\}$ for each i is the Prüfer code of exactly one labeled n-vertex tree.

Proof. **Uniqueness**. By (P1) we know $a_{n-1} = n$. Then by (P3), we can reconstruct b_i for all $1 \le i \le n-1$. Thus the edges are $a_1 b_1, \dots, a_{n-1} b_{n-1}$.

Existence. Given $(a_1, \dots, a_{n-2})$, we let $a_{n-1} = n$ and define numbers b_i by (P3). Now consider the edges going from $a_{n-1} b_{n-1}$ backwards and check that for each i, b_i is a leaf in the graph formed by the edges $a_i b_i, \dots, a_{n-1} b_{n-1}$. □

Theorem 2.4 (Prüfer, 1918): Every vector $(a_1, \dots, a_{n-2})$ with $a_i \in \{1, \dots, n\}$ for each i is the Prüfer code of exactly one labeled n-vertex tree.

Proof. **Uniqueness**. By (P1) we know $a_{n-1} = n$. Then by (P3), we can reconstruct b_i for all $1 \leq i \leq n-1$. Thus the edges are $a_1 b_1, \dots, a_{n-1} b_{n-1}$.

Existence. Given $(a_1, \dots, a_{n-2})$, we let $a_{n-1} = n$ and define numbers b_i by (P3). Now consider the edges going from $a_{n-1} b_{n-1}$ backwards and check that for each i, b_i is a leaf in the graph formed by the edges $a_i b_i, \dots, a_{n-1} b_{n-1}$. □

Corollary 2.5 (Cayley's Formula, Borchardt 1860): There are n^{n-2} labeled n-vertex trees.