Chapter 2

수형도 Trees

이 단원에서는 수형도 trees에 대해 공부하겠습니다. 정의를 상기해봅시다.

A tree is a connected simple graph with no cycle.

2.1 Characterizations

Lemma 2.1. *If T is a tree with $|V| \geq 2$ then it contains at least two end vertices.*

Proof. All paths have length $\leq |V| - 1$ (upper bound). Hence we can find a longest path. Claim: the initial and final vertices are end vertices. □

Lemma 2.2. *If T is a tree with $|V| \geq 2$ then there is a unique path from v_i to v_j.*

Proof. Since T is connected, there is at least one path from v_i to v_j. Suppose there are two paths. Then you can find a cycle, which is a contradiction. □

Lemma 2.3. *If T is a tree with $|V| \geq 2$ then $|E| = |V| - 1$.*

Proof. (Induction on the number of vertices) First, if $|V| = 2$ then $|E| = 1$. This is because T should be connected and has no loops or multiple edges. Let us assume that for all threes with $2 \geq |V| < m$, $|E| = |V| - 1$, and let T be a tree with $|V| = m$. Choose an end vertex v of T. Then, $T - v$ is now a tree with $m - 1$ vertices. By induction hypothesis, $T - v$ has $m - 2$ edges. This shows that T has $m - 1$ edges. □

Theorem 2.4. *Let T be a simple graph with $|V| = p$. The following are equivalent.*

1. *T is a tree.*
2. *T has $p-1$ edges and no cycles.*
3. *T has $p-1$ edges and it is connected.*

Proof. See Anderson Theorem 3.4, p48. □

2.2 Cayley's formula

먼저 필요한 용어에 대한 정의를 소개합니다.

1. For a graph $G = (V, E)$, a vertex labeling is a function from V to a set of labels. A graph with a vertex labeling is called a vertex-labeled graph.
2. Similarly, an edge labeling is a function from E to a set of labels. A graph with an edge labeling is called an edge-labeled graph.
3. If real numbers are used to label edges, then the associated edge-labeled graph is often called a weighted graph.
4. Without any qualification, a labeled graph is (usually) a vertex-labeled graph with vertext labeling given by a bijection
$$f : V \longrightarrow \{1, 2, ..., |V|\}.$$

Theorem 2.5 (Cayley's formula)**.** *For every positive integer n, the number of trees on n labeled vertices is*
$$n^{n-2}.$$

2.2.1 Bijective proof: Prüfer's proof (1918)

다음과 같은 일대일 대응 함수를 구성한 후, 집합 Y의 크기를 계산하여 공식을 얻을 수 있습니다.

$$\phi : X = \{\text{all labeled trees with } n \text{ vertices}\} \to Y = \{(a_1, ..., a_{n-2}) : 1 \le a_1, a_2, ..., a_{n-2} \le n\}$$

1. For a tree, $T \in X$, find the end vertex v in T with the smallest label.

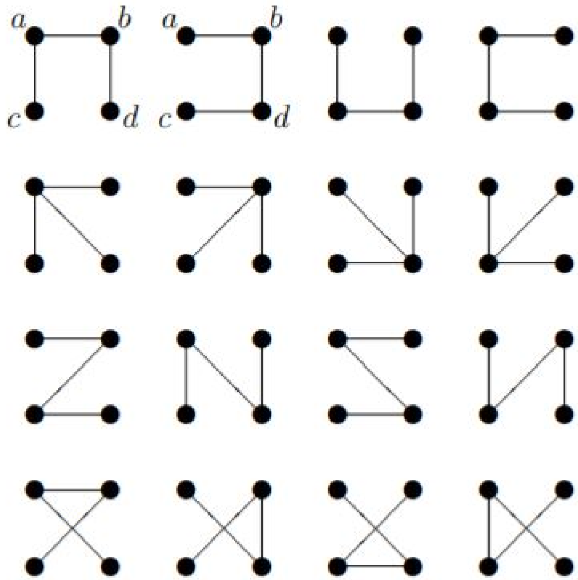

Figure 2.1: Cayley's formula ($n = 4$).

2. Record the label a_1 of the vertex adjacent to v, and then remove v.

3. Find the end vertext v' in $T - v$ with the smallest label.

4. Record the label a_2 of the vertex adjacent to v', and then remove v'.

5. Continue this process until you have only two vertices. Note that one of these two vertices is labeled by n (why?), and thus $a_{n-1} = n$.

6. Note that the degree of a vertex v labeled by k is equal to

$$deg(v) = 1 + \text{the number of times } k \text{ appearing in the corresponding sequence.}$$

7. To see the inverse of ϕ, for each sequence $(a_1, ..., a_{n-2})$ compute

$$\begin{aligned} b_1 &= \min\{1, 2, ..., n\} \setminus \{a_1, ..., a_{n-2}\} \\ b_2 &= \min\{1, 2, ..., n\} \setminus \{a_2, ..., a_{n-2}, b_1\} \\ b_3 &= \min\{1, 2, ..., n\} \setminus \{a_3, ..., a_{n-2}, b_1, b_2\} \\ &\vdots \\ b_{n-1} &= \min\{1, 2, ..., n\} \setminus \{b_1, ..., b_{n-2}\} \end{aligned}$$

For each $1 \leq i \leq n-1$, connect the vertices labeled by a_i and b_i with an edge.

2.2.2 Mathematical induction: Riordan's proof (1968)

1. Let $T_{n,k}$ be the number of labeled forests on n vertices with k trees such that vertices $1, 2, ..., k$ belong to different trees.

$$1\bullet \quad 2\bullet \quad 3\bullet \quad \cdots \quad \bullet\, k-1 \quad \bullet\, k$$

2. We set $T_{0,0} = 1$ and $T_{n,0} = 0$ for $n > 0$. Note that $T_{0,0} = 1$ is necessary so that $T_{n,n} = 1$.

3. Vertex 1 can be adjacent to i of the remaining $n-k$ vertices $\{k+1, ..., n\}$ for $0 \leq i \leq n-k$.

4. If we delete vertex 1 from the forest, then
   - we have a forest with $n-1$ vertices,
   - we have $k-1+i$ vertices ($\{2, 3, ..., k\}$ together with i vertices chosen in the above) that must be in separate trees (why?).

5. This shows that $T_{n,k}$ satisfies

$$T_{n,k} = \sum_{i=0}^{n-k} \binom{n-k}{i} T_{n-1,k-1+i}.$$

6. Now using mathematical induction with the above identity to show that

$$T_{n,k} = kn^{n-k-1}.$$

7. Now if $k = 1$ then we obtain Cayley's formula.

2.2.3 Generating function method: Shukla's proof (2018)

1. Let $T(n)$ be the set of all labeled trees with n vertices and $E_{12} = E_{12}(n)$ be the set of all trees in $T(n)$ which contain the edge 12 incident with vertices 1 and 2.

$$1 \bullet\!\!-\!\!\!-\!\!\!-\!\!\bullet 2$$

2. Note that $|E_{12}| = |E_{ij}|$ for any edge ij.

3. Then the number of possible edges in all the trees in $T(n)$ is

$$\binom{n}{2} \cdot |E_{12}| = (n-1) \cdot |T(n)| \tag{2.1}$$

(note that each tree in $T(n)$ has $(n-1)$ edges).

4. For each $T \in E_{12}$, once the edge 12 is removed, $T - 12$ is a forest having two trees. Assume that there are k vertices (for $0 \le k \le n-2$) on the left hand side of vertex 1 and $n-k-2$ vertices on the right hand side of vertex 2.

   (a tree with k vertices) ——— $\bullet$ 1 2 $\bullet$ ——— (a tree with $n-k-2$ vertices)

   Then with vertices 1 and 2, they form trees with $k+1$ vertices and $n-k-1$ vertices respectively.

5. This shows that
$$|E_{12}| = \sum_{k=0}^{n-2} \binom{n-2}{k} |T(k+1)| \cdot |T(n-k-1)|. \tag{2.2}$$

6. Combine Eq. (2.1) and Eq. (2.2) to obtain
$$|T(n)| = \frac{n}{2} \sum_{k=0}^{n-2} \binom{n-2}{k} |T(k+1)| \cdot |T(n-k-1)|.$$

7. Now using the above identity, show that the exponential generating function
$$g(x) = \sum_{n \ge 1} |T(n)| \frac{x^n}{(n-1)!}$$
   satisfies the identity
$$g(x) = \ln \frac{g(x)}{x}$$
   and then apply the Lagrange inversion formula to conclude that $|T(n)| = n^{n-2}$.

2.2.4 Cayley's original proof (1889)

가능한 labeled tree T 하나에 다음과 같은 단항식을 대응시켜봅시다.

$$x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$$

여기서 d_i는 T가 갖고 있는 vertex v_i의 차수degree입니다. 그리고 아래 다변수 다항식의 전개를 살펴봅시다.

1. $n = 2$일때
$$(x_1 + x_2)^0 (x_1 x_2) = x_1 x_2.$$

2. $n = 3$일때
$$(x_1 + x_2 + x_3)^1 (x_1 x_2 x_3) = x_1^2 x_2 x_3 + x_1 x_2^2 x_3 + x_1 x_2 x_3^2.$$

3. $n = 4$일때

$$
\begin{aligned}
&(x_1 + x_2 + x_3 + x_4)^2(x_1x_2x_3x_4) \\
=& (x_1^2 + 2x_1x_2 + 2x_1x_3 + 2x_1x_4 + x_2^2 + 2x_2x_3 + 2x_2x_4 + x_3^2 + 2x_3x_4 + x_4^2)(x_1x_2x_3x_4)
\end{aligned}
$$

2.2.5 Principle of inclusion-exclusion

J.W. Moon's proof. See Anderson §6.3; Cioba and Ram Mutty (2nd edition) §5.2.

2.2.6 Double counting

Jim Pitman's proof. See Proofs from THE Book (5th edition), §33.

2.2.7 Kirchhoff's matrix-tree theorem

See Proofs from THE Book (5th edition), §33.

2.3 Spanning trees

A spanning tree of a connected graph G is a subgraph of G which is a tree and has all the vertices of G.

Theorem 2.6. *Every connected graph G has a spanning tree.*

Proof.
1. We can assume that G is simple (why?). Induction on the number of cycles.
2. If G has no cycles then it is a tree and thus G itself is its spanning tree.
3. Assume that every G with less than n cycles has a spanning tree.
4. If G has n cycles then we can remove an edge e in a cycle so that $G - e$ is still connected and has $n - 1$ cycles. By the induction hypothesis, $G - e$ has a spanning tree T.
5. Note that T is also a spanning tree of G.

□

Proposition 2.7. *Let $\tau(G)$ denote the number of spanning trees of a connected graph G. If e is an edge of G such that... then*

$$\tau(G) = \tau(G - e) + \tau(G/e).$$

Proposition 2.8. *The number of spanning trees of a labeled complete graph K_n is n^{n-2}.*

Proof. Use Cayley's formula. ☐

Theorem 2.9 (Kruskal's algorithm)**.** *For a weighted graph G, the following algorithm provides a spanning tree of G such that the total weight of all the edges in the tree is minimized.*

1. *Choose an edge having the smallest weight.*
2. *Choose from the edges not chosen yet, choose the edge with the smallest weight which will not create a cycle.*
3. *Continue until you get a spanning tree.*

Example 2.10. *Apply Kruskal's algorithem to the weighted graph in Figure 2.2.*

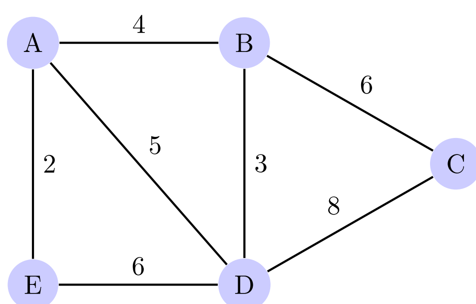

Figure 2.2: Kruskal's algorithm

2.4 Parking function

Definition 2.11 (Parking functions)**.** *A parking function of size n is a sequence $(a_1, a_2, ..., a_n)$ of positive integers such that if $b_1 \leq b_2 \leq \cdots \leq b_n$ is the non-decreasing rearrangement of $a_1, ..., a_n$ then $b_i \leq i$ for all i.*

The number of parking functions of size n is $(n+1)^{n-1}$.