

Elementary Introduction to Graph Theory

(01YEIG 2026/2027)

Lecture 4



Francesco Dolce
dolcefra@cvut.cz

October 16, 2026

updated: October 16, 2026

PDF available at the address:

dolcefra.pages.fit/ens/2627/YEIG-lecture-04.pdf

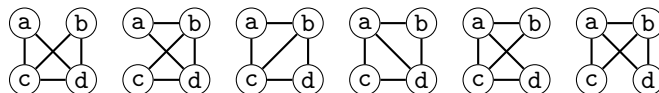
Solutions of Exercises in the previous Lecture

Solution of Exercise $[S_4]$. The number of connected graphs on 4 vertices, S_4 , is 38. These 38 graphs are:

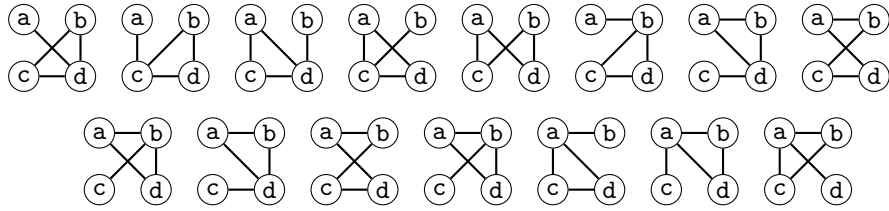
a) 1 complete graph;



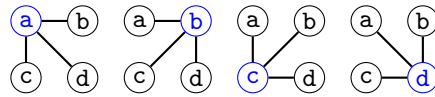
b) $6 = \binom{6}{1}$ graphs with one edge missing;



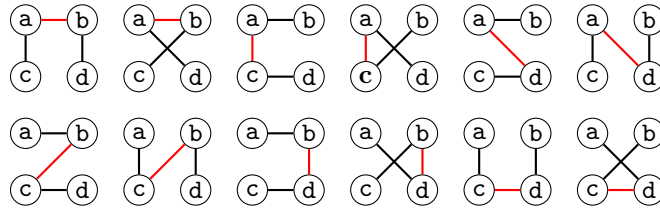
c) $15 = \binom{6}{2}$ graphs with two edges missing;



- d) 4 graphs with **one vertex** with degree 3, connected to the other 3 vertices of degree 1;

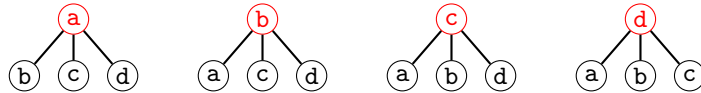


- e) $12 = 6 \cdot 2$ graphs that are paths of length 3, since we have 6 possibilities for the **middle segment** and 2 possibilities for the end segments;

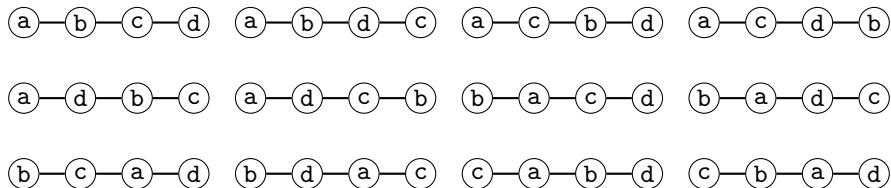


Solution of Exercise [Cayley's formula]. The $4^{4-2} = 16$ labelled trees on the set of vertices $\{a, b, c, d\}$ can be divided into two sets:

- a) 4 trees isomorphic to $K_{1,3}$, one for each possible choice of the **centre**;



- b) $12 = \frac{4!}{2}$ trees isomorphic to P_4 , one for each permutation of the vertices up to reversal (reversing the permutation gives the same path);



1 Rooted and ordered trees

A *rooted tree* $T(r)$ is a tree $T = (V, E)$ with a specified vertex $r \in V$ called the *root* of T . Given a vertex v in a rooted tree $T(r)$ different from r , we call the vertex in $N_T(v)$ on the unique path between r and v a *parent* of v ; all vertices in this path are called *ancestors* of v . Similarly, every non-root vertex is a *child* of its parent and u is a *descendant* of v if v is an ancestor of u .

The *level* $\ell_T(v)$ of a vertex v in a rooted tree $T(r)$ is defined as the distance $d_T(r, v)$ (i.e., the length of the path from r to v in the tree). The root is the only vertex with level 0. The *height* of a rooted tree is the maximal level.

When dealing with rooted trees, we call *leaf* a vertex that has no children. Hence, in a non-trivial rooted tree the root is not a leaf (in the trivial rooted tree, we have only one vertex that is at the same time the root and a leaf).

Example 1 The graph in Figure 1 is a rooted tree with root r . The vertices a, b, c are children of r . The vertex h is a child of e and a descendant of a (hence e is a parent of h and a an ancestor of h). The vertices d, e, f, g have level 2, while the height of the tree is 3 (h is a vertex of maximal level).

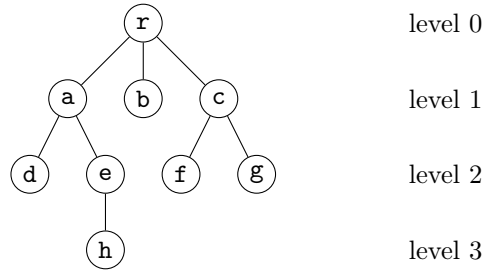


Figure 1: A rooted tree.

An *ordered tree* $T(r, \preceq)$ is a rooted tree $T(r) = (V, E)$ with root r and partial order $\preceq$ on V giving, for each vertex v , a total order on the children of v (i.e., we don't compare vertices that are not siblings). We usually represent it by drawing the children of a node from left to right starting from the smallest.

Example 2 Let T be the rooted tree in Example 1 and represented in Figure 1. Then, $T(r, \preceq)$, with the partial order $\preceq$ defined by

$$a \prec b \prec c, \quad d \prec e \quad \text{and} \quad f \prec g$$

is an ordered tree.

2 Adjacency matrix

Let $G = (V, E)$ be a simple graph with $V = \{v_1, v_2, \dots, v_n\}$. The *adjacency matrix* A_G is the $n \times n$ matrix defined by

$$[A_G]_{i,j} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \in E \\ 0 & \text{otherwise} \end{cases}.$$

Clearly A_G is a non-negative symmetric matrix.

Example 3 Let $G = (\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}, \{\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{a}, \mathbf{c}\}, \{\mathbf{a}, \mathbf{d}\}, \{\mathbf{b}, \mathbf{c}\}\})$. The adjacency matrix of G is given by

$$A_G = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Note that the definition of A_G can be generalised to non-simple graphs by replacing 1 with the number of parallel edges and counting each loop once.

Theorem 1 Let $k \in \mathbb{N}_0$ and let A_G be the adjacency matrix of a graph $G = (V, E)$ with $V = \{v_1, v_2, \dots, v_n\}$. Then $[A_G^k]_{i,j}$ is the number of walks of length k with endpoints v_i and v_j in G .

Proof. The case $k = 0$ is trivial since we have $A_G^0 = I_n$, and the only walks of length 0 are the ones connecting a vertex to itself.

Let us prove it by induction on $k \geq 1$.

($k = 1$) It follows directly from the definition of adjacency matrix, since a walk of length 1 is exactly an edge.

($k - 1 \rightarrow k$) The (i, j) -element of A_G^k is given by

$$[A_G^k]_{i,j} = [A_G^{k-1} \cdot A_G]_{i,j} = \sum_{\ell=1}^n [A_G^{k-1}]_{i,\ell} \cdot [A_G]_{\ell,j} = \sum_{\substack{\ell=1 \\ \{v_\ell, v_j\} \in E}}^n [A_G^{k-1}]_{i,\ell}$$

but $[A_G^{k-1}]_{i,\ell}$ is, by induction hypothesis, the number of walks of length $k - 1$ with endpoints v_i and v_ℓ . Since we are summing over all walks that can be prolonged from v_ℓ to v_j (for every possible choice of v_ℓ) we can conclude (see Figure 2).

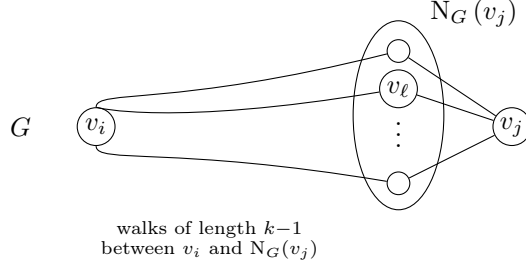


Figure 2: Illustration of the proof of Theorem 1. ■

Recall that a matrix is *positive* if all its entries are strictly positive.

Corollary 1 *A graph $G = (V, E)$ is connected if and only if $\sum_{k=0}^{\#V-1} A_G^k$ is a positive matrix.*

Proof.

($\Rightarrow$) Let $v_i, v_j \in V$. Since G is connected the two vertices are linked in G . Thus, there is a walk of length $k \leq \#V - 1$ in G with endpoints v_i and v_j . Indeed, a shortest path between two vertices is simple and therefore contains at most $\#V - 1$ edges. Hence, $[A_G^k]_{i,j} \geq 1$.

($\Leftarrow$) Let i, j with $1 \leq i, j \leq n$. Since $\left[\sum_{k=0}^{n-1} A_G^k \right]_{i,j} > 0$, there exists ℓ , with $0 \leq \ell \leq n - 1$ such that $[A_G^\ell]_{i,j} \geq 1$. Thus, there is a walk of length ℓ in G with endpoints v_i and v_j . ■

Example 4 Let A_G be the adjacency matrix seen in Example 3. We have

$$A_G^2 = \begin{pmatrix} \textcolor{red}{3} & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad A_G^3 = \begin{pmatrix} 2 & 4 & \textcolor{blue}{4} & 3 \\ 4 & 3 & 3 & 2 \\ 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}.$$

Indeed, we have, e.g., **three** walks of length 2 from **a** to **a**, namely

$$(\mathbf{a}, \mathbf{b}, \mathbf{a}), (\mathbf{a}, \mathbf{c}, \mathbf{a}) \quad \text{and} \quad (\mathbf{a}, \mathbf{d}, \mathbf{a});$$

and **four** walks of length 3 from **a** to **c**, namely:

$$(\mathbf{a}, \mathbf{b}, \mathbf{a}, \mathbf{c}), (\mathbf{a}, \mathbf{c}, \mathbf{a}, \mathbf{c}), (\mathbf{a}, \mathbf{c}, \mathbf{b}, \mathbf{c}) \quad \text{and} \quad (\mathbf{a}, \mathbf{d}, \mathbf{a}, \mathbf{c}).$$

Notice also that

$$\sum_{k=0}^3 A_G^k = I_4 + A_G + A_G^2 + A_G^3 = \begin{pmatrix} 6 & 6 & 6 & 4 \\ 6 & 6 & 5 & 3 \\ 6 & 5 & 5 & 2 \\ 4 & 3 & 2 & 2 \end{pmatrix}$$

is a positive matrix.

3 Isomorphism of graphs

Two graphs $G = (V, E)$ and $H = (U, F)$ are *identical* if $V = U$ and $E = F$.

They are *isomorphic*, denoted $G \cong H$, if there exists a bijection $\varphi : V \rightarrow U$, called an *isomorphism*, such that for every $u, v \in V$

$$\{u, v\} \in E \Leftrightarrow \{\varphi(u), \varphi(v)\} \in F.$$

Clearly, when $\varphi = \text{id}$, the two graphs are identical.

Example 5 The graphs $G = \left(V, \binom{V}{2}\right)$ and $H = \left(U, \binom{U}{2}\right)$ with $V = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$ and $U = \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t}\}$ are isomorphic (see Figure 3), with bijection, e.g.,

$$\varphi : \begin{cases} \mathbf{a} \mapsto \mathbf{x} \\ \mathbf{b} \mapsto \mathbf{y} \\ \mathbf{c} \mapsto \mathbf{z} \\ \mathbf{d} \mapsto \mathbf{t} \end{cases}$$

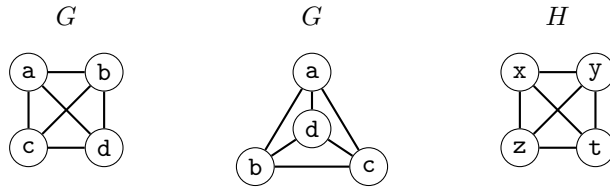


Figure 3: Three isomorphic graphs (the first two are identical).

Thus, two graphs are isomorphic if they are "the same" up to relabelling the vertices. Clearly, degree and distance are preserved under isomorphism.

Proposition 1 Let $G = (V, E)$ and $H = (U, F)$ be two isomorphic graphs with isomorphism φ . Then, for every $u, v \in V$ we have

1. $d_G(v) = d_H(\varphi(v))$,
2. $d_G(u, v) = d_H(\varphi(u), \varphi(v))$.

Proof.

1. From the definition of isomorphism it follows that $u \in N_G(v)$ if and only if $\varphi(u) \in N_H(\varphi(v))$. Thus, the cardinality of the two sets, i.e., the two degrees $d_G(v)$ and $d_H(\varphi(v))$, coincide.
2. Exercise.
(Hint: show that φ maps every path in G to a path of the same length in H .)

■

Example 6 The graphs represented in Figure 4 are isomorphic. A possible bijection is

$$\varphi : \begin{cases} a \mapsto 1 \\ b \mapsto 4 \\ c \mapsto 2 \\ d \mapsto 5 \\ e \mapsto 3 \\ f \mapsto 6 \end{cases} .$$

One can check, e.g., that $d_G(a) = d_H(1) = 3$, and that $d_G(a, e) = d_H(1, 3) = 2$.



Figure 4: Two isomorphic graphs.

We normally do not distinguish between isomorphic graphs. When dealing with a graph up to isomorphism, we can forget the labels in the drawing.

If two graphs are isomorphic, then clearly they have the same order and the same size. The converse is false.

Exercise [Non-isomorphic graphs]. Show that the two unlabelled graphs in Figure 5, which have both order 4 and size 3, are not isomorphic.



Figure 5: Two non-isomorphic graphs with the same order and size.

The graph isomorphism problem, i.e., deciding whether two graphs are isomorphic, belongs to $\mathcal{NP}$. It is not known whether it belongs to $\mathcal{P}$ or is $\mathcal{NP}$ -complete.

4 Tree isomorphism problem

We will see a polynomial-time algorithm for the isomorphism of trees. We proceed in three steps, obtaining three different algorithms:

1. an algorithm for isomorphism of ordered trees;
2. an algorithm for isomorphism of rooted trees;
3. a general algorithm for the isomorphism of (unrooted) trees.

Two trees $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ are isomorphic, denoted

$$T_1 \cong T_2,$$

if the two trees are isomorphic as graphs, i.e., there exists a bijective mapping $\varphi : V_1 \rightarrow V_2$ such that for every $u, v \in V_1$

$$\{u, v\} \in E_1 \iff \{\varphi(u), \varphi(v)\} \in E_2.$$

Two rooted trees $T_1(r_1)$ and $T_2(r_2)$ are isomorphic, denoted

$$T_1(r_1) \cong_R T_2(r_2),$$

if $T_1 \cong T_2$ with isomorphism φ , with the root sent to the root, i.e., $\varphi(r_1) = r_2$.

Two ordered trees $T_1(r_1, \preceq_1)$ and $T_2(r_2, \preceq_2)$ are isomorphic, denoted

$$T_1(r_1, \preceq_1) \cong_O T_2(r_2, \preceq_2),$$

if $T_1(r_1) \cong_R T_2(r_2)$ with isomorphism φ , and the mapping φ preserves the order of the children of each vertex, i.e., for every $u, v \in V_1$ having the same parent we have

$$u \preceq_1 v \iff \varphi(u) \preceq_2 \varphi(v).$$

Example 7 The two trees T_1 and T_2 in Figure 6 are isomorphic. Indeed, an isomorphism is given by

$$\varphi : \begin{cases} \mathbf{a} \mapsto \mathbf{E} \\ \mathbf{b} \mapsto \mathbf{D} \\ \mathbf{c} \mapsto \mathbf{C} \\ \mathbf{d} \mapsto \mathbf{B} \\ \mathbf{e} \mapsto \mathbf{A} \end{cases}.$$

However, as rooted trees, $T_1(\mathbf{d})$ and $T_2(\mathbf{C})$ are not isomorphic since there is no isomorphism sending $\mathbf{d}$ to $\mathbf{C}$ (why?)

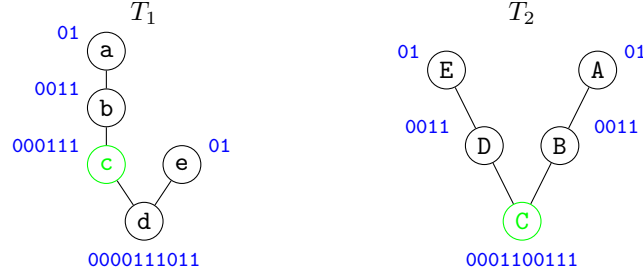


Figure 6: $T_1 \cong T_2$ but $T_1(d) \not\cong_R T_2(c)$.

Example 8 The two rooted trees $T_3(r)$ and $T_4(R)$ in Figure 7 are isomorphic. Indeed, it is enough to consider the bijection

$$\varphi : \begin{cases} r \mapsto R \\ a \mapsto C \\ b \mapsto B \\ c \mapsto D \\ d \mapsto A \end{cases} .$$

They are not isomorphic as ordered trees (why?)

The two ordered trees $T_4(R, \preceq_4)$ and $T_5(s, \preceq_5)$ in the same figure are isomorphic (find the isomorphism!).

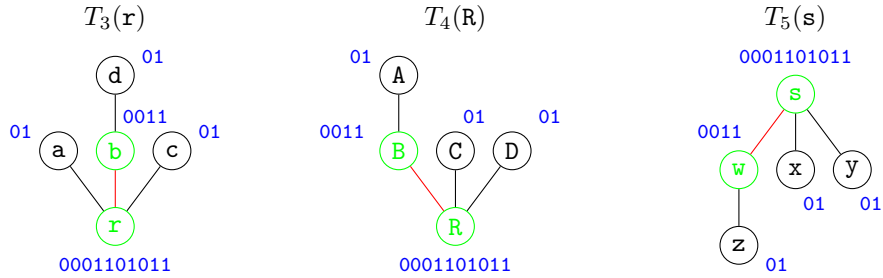


Figure 7: Three ordered trees.

5 Isomorphism of ordered trees

We will encode an ordered tree by a sequence of 0s and 1s such that:

- the encoding uses only properties fixed by the mapping φ ;

- the code is uniquely (up to isomorphism) decodable.

These two properties of the encoding imply that two ordered trees are isomorphic if and only if their codes are the same.

We code an ordered tree $T(r, \preceq)$ in the following way:

- given a node v with children $v_1 \prec v_2 \prec \dots \prec v_k$ having encoding $C(v_1), C(v_2), \dots, C(v_k)$, we define $C(v) = 0 C(v_1) C(v_2) \dots C(v_k) 1$;
- $C(T) = C(r)$, i.e., the code of a tree coincides with the code of its root.

In particular, since a leaf v has no children, we have $C(v) = 01$.

Example 9 Let $T_3(\mathbf{r}, \preceq_3)$ be the ordered tree in Example 8. Then

$$C(\mathbf{a}) = C(\mathbf{c}) = C(\mathbf{d}) = 01; \quad C(\mathbf{b}) = 0.C(\mathbf{d}).1 = 0011;$$

and

$$C(T_3) = C(\mathbf{r}) = 0.C(\mathbf{a}).C(\mathbf{b}).C(\mathbf{c}).1 = 0010011011.$$

One can check that $C(T_4) = C(T_5) = 0001101011 \neq C(T_3)$.

Every code constructed this way contains the same number of 0s and 1s. Moreover, the number of 0s is strictly greater than the number of 1s in every proper prefix.

We can also decode a code C of an ordered tree $T(r, \prec)$. If $C(T) = 0w1$, then w is the concatenation of the codes of the ordered subtrees rooted at the children of r .

In particular, if $w = C(v_1)C(v_2) \dots C(v_k)$, then $C(v_1)$ is the shortest non-empty prefix of w with the same number of 0s and 1s. We can "cut" $C(v_1)$ and continue analogously.

Example 10 Let $C(T_3) = 0.01001101.1$ be as in Example 9. The shortest prefix of $w = 01001101$ having the same number of 0s and 1s is $C(v_1) = 01$. Similarly $C(v_2) = 0011$ and $C(v_3) = 01$. The vertex v_2 has a child v_4 with encoding $C(v_4) = 01$. Thus, v_1, v_3, v_4 are leaves and T_3 is isomorphic to the ordered tree $T = (\{r, v_1, v_2, v_3, v_4\}, \{\{r, v_1\}, \{r, v_2\}, \{r, v_3\}, \{v_2, v_4\}\})$.