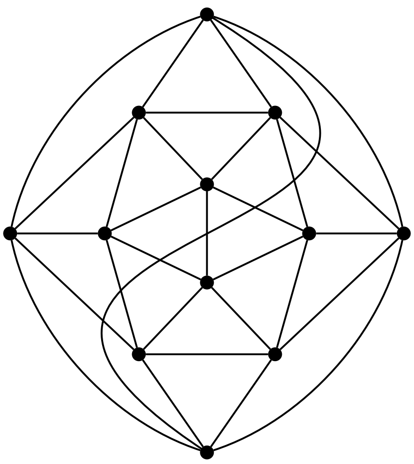
Lecture Notes on

GRAPH THEORY

Tero Harju
Department of Mathematics
University of Turku
FIN-20014 Turku, Finland
e-mail: harju@utu.fi
1994 – 2011

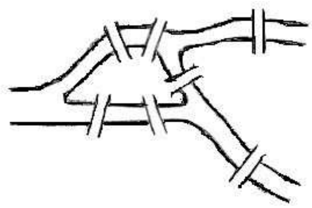


Contents

1

Introduction

Graph theory may be said to have its beginning in 1736 when EULER considered the (general case of the) **Königsberg bridge problem**: Does there exist a walk crossing each of the seven bridges of Königsberg exactly once? (Solutio Problematis ad geometriam situs pertinentis, *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 8 (1736), pp. 128-140.)



It took 200 years before the first book on graph theory was written. This was "Theorie der endlichen und unendlichen Graphen" (Teubner, Leipzig, 1936) by KÖNIG in 1936. Since then graph theory has developed into an extensive and popular branch of mathematics, which has been applied to many problems in mathematics, computer science, and other scientific and not-so-scientific areas. For the history of early graph theory, see

N.L. BIGGS, R.J. LLOYD AND R.J. WILSON, "Graph Theory 1736 – 1936", Clarendon Press, 1986.

There are no standard notations for graph theoretical objects. This is natural, because the names one uses for the objects reflect the applications. Thus, for instance, if we consider a communications network (say, for email) as a graph, then the computers taking part in this network, are called nodes rather than vertices or points. On the other hand, other names are used for molecular structures in chemistry, flow charts in programming, human relations in social sciences, and so on.

These lectures study *finite graphs* and majority of the topics is included in

J.A. BONDY, U.S.R. MURTY, "Graph Theory with Applications", Macmillan, 1978.

R. DIESTEL, "Graph Theory", Springer-Verlag, 1997.

F. HARARY, "Graph Theory", Addison-Wesley, 1969.

D.B. WEST, "Introduction to Graph Theory", Prentice Hall, 1996.

R.J. WILSON, "Introduction to Graph Theory", Longman, (3rd ed.) 1985.

In these lectures we study *combinatorial aspects* of graphs. For more *algebraic* topics and methods, see

N. BIGGS, "Algebraic Graph Theory", Cambridge University Press, (2nd ed.) 1993.

C. GODSIL, G.F. ROYLE, "Algebraic Graph Theory", Springer, 2001.
and for *computational aspects*, see

S. EVEN, "Graph Algorithms", Computer Science Press, 1979.

In these lecture notes we mention several open problems that have gained respect among the researchers. Indeed, graph theory has the advantage that it contains easily formulated open problems that can be stated early in the theory. Finding a solution to any one of these problems is another matter.

Sections with a star ($*$) in their heading are optional.

Notations and notions

• For a finite set X, $|X|$ denotes its size (cardinality, the number of its elements).
• Let
$$[1,n] = \{1,2,\dots,n\},$$
and in general,
$$[i,n] = \{i,i+1,\dots,n\}$$
for integers $i \leq n$.
• For a real number x, the **floor** and the **ceiling** of x are the integers
$$\lfloor x \rfloor = \max\{k \in \mathbb{Z} \mid k \leq x\} \text{ and } \lceil x \rceil = \min\{k \in \mathbb{Z} \mid x \leq k\}.$$
• A family $\{X_1, X_2, \dots, X_k\}$ of subsets $X_i \subseteq X$ of a set X is a **partition** of X, if
$$X = \bigcup_{i \in [1,k]} X_i \quad \text{and} \quad X_i \cap X_j = \emptyset \text{ for all different } i \text{ and } j\,.$$
• For two sets X and Y,
$$X \times Y = \{(x,y) \mid x \in X,\ y \in Y\}$$
is their **Cartesian product**, and
$$X \triangle Y = (X \setminus Y) \cup (Y \setminus X)$$
is their **symmetric difference**. Here $X \setminus Y = \{x \mid x \in X, x \notin Y\}$.
• Two integers $n,k \in \mathbb{N}$ (often $n = |X|$ and $k = |Y|$ for sets X and Y) have the **same parity**, if both are even, or both are odd, that is, if $n \equiv k \pmod 2$. Otherwise, they have opposite parity.

Graph theory has abundant examples of **NP-complete problems**. Intuitively, a problem is in P [1] if there is an efficient (practical) algorithm to find a solution to it. On the other hand, a problem is in NP [2], if it is first efficient to guess a solution and then efficient to check that this solution is correct. It is conjectured (and not known) that P $\neq$ NP. This is one of the great problems in modern mathematics and theoretical computer science. If the guessing in NP-problems can be replaced by an efficient systematic search for a solution, then P=NP. For any one NP-complete problem, if it is in P, then necessarily P=NP.

[1] Solvable – by an algorithm – in polynomially many steps on the size of the problem instances.
[2] Solvable *nondeterministically* in polynomially many steps on the size of the problem instances.

1.1 Graphs and their plane figures

Let V be a *finite* set, and denote by

$$E(V) = \{\{u, v\} \mid u, v \in V,\ u \neq v\}\,.$$

the **2-sets** of V, i.e., subsets of two distinct elements.

DEFINITION. A pair $G = (V, E)$ with $E \subseteq E(V)$ is called a **graph** (**on** V). The elements of V are the **vertices** of G, and those of E the **edges** of G. The vertex set of a graph G is denoted by V_G and its edge set by E_G. Therefore $G = (V_G, E_G)$.

In literature, graphs are also called *simple graphs*; vertices are called *nodes* or *points*; edges are called *lines* or *links*. The list of alternatives is long (but still finite).

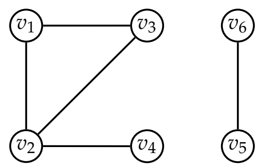
A pair $\{u, v\}$ is usually written simply as uv. Notice that then $uv = vu$. In order to simplify notations, we also write $v \in G$ and $e \in G$ instead of $v \in V_G$ and $e \in E_G$.

DEFINITION. For a graph G, we denote

$$\nu_G = |V_G| \text{ and } \varepsilon_G = |E_G|\,.$$

The number ν_G of the vertices is called the **order** of G, and ε_G is the **size** of G. For an edge $e = uv \in G$, the vertices u and v are its **ends**. Vertices u and v are **adjacent** or **neighbours**, if $uv \in G$. Two edges $e_1 = uv$ and $e_2 = uw$ having a common end, are **adjacent** with each other.

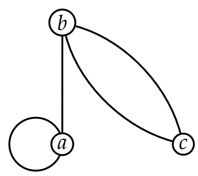
A graph G can be represented as a plane figure by drawing a line (or a curve) between the points u and v (representing vertices) if $e = uv$ is an edge of G. The figure on the right is a geometric representation of the graph G with $V_G = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and $E_G = \{v_1v_2, v_1v_3, v_2v_3, v_2v_4, v_5v_6\}$.



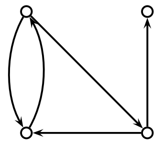
Often we shall omit the identities (names v) of the vertices in our figures, in which case the vertices are drawn as anonymous circles.

Graphs can be generalized by allowing **loops** vv and **parallel** (or **multiple**) **edges** between vertices to obtain a **multigraph** $G = (V, E, \psi)$, where $E = \{e_1, e_2, \dots, e_m\}$ is a set (of symbols), and $\psi\colon E \to E(V) \cup \{vv \mid v \in V\}$ is a function that attaches an unordered pair of vertices to each $e \in E$: $\psi(e) = uv$.

Note that we can have $\psi(e_1) = \psi(e_2)$. This is drawn in the figure of G by placing two (parallel) edges that connect the common ends. On the right there is (a drawing of) a multigraph G with vertices $V = \{a, b, c\}$ and edges $\psi(e_1) = aa$, $\psi(e_2) = ab$, $\psi(e_3) = bc$, and $\psi(e_4) = bc$.



Later we concentrate on (simple) graphs.



DEFINITION. We also study **directed graphs** or **digraphs** $D = (V, E)$, where the edges have a direction, that is, the edges are ordered: $E \subseteq V \times V$. In this case, $uv \neq vu$.

The directed graphs have representations, where the edges are drawn as arrows. A digraph can contain edges uv and vu of *opposite directions*.

Graphs and digraphs can also be coloured, labelled, and weighted:

DEFINITION. A function $\alpha \colon V_G \to K$ is a **vertex colouring** of G by a set K of colours. A function $\alpha \colon E_G \to K$ is an **edge colouring** of G. Usually, $K = [1, k]$ for some $k \geq 1$.

If $K \subseteq \mathbb{R}$ (often $K \subseteq \mathbb{N}$), then α is a **weight function** or a **distance function**.

Isomorphism of graphs

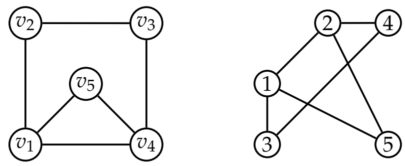
DEFINITION. Two graphs G and H are **isomorphic**, denoted by $G \cong H$, if there exists a bijection $\alpha \colon V_G \to V_H$ such that

$$uv \in E_G \iff \alpha(u)\alpha(v) \in E_H$$

for all $u, v \in G$.

Hence G and H are isomorphic if the vertices of H are renamings of those of G. Two isomorphic graphs enjoy the same graph theoretical properties, and *they are often identified*. In particular, all isomorphic graphs have the same plane figures (excepting the identities of the vertices). This shows in the figures, where we tend to replace the vertices by small circles, and talk of 'the graph' although there are, in fact, infinitely many such graphs.

Example 1.1. The following graphs are isomorphic. Indeed, the required isomorphism is given by $v_1 \mapsto 1$, $v_2 \mapsto 3$, $v_3 \mapsto 4$, $v_4 \mapsto 2$, $v_5 \mapsto 5$.



Isomorphism Problem. *Does there exist an efficient algorithm to check whether any two given graphs are isomorphic or not?*

The following table lists the number $2^{\binom{n}{2}}$ of all graphs on a given set of n vertices, and the number of all nonisomorphic graphs on n vertices. It tells that at least for computational purposes an efficient algorithm for checking whether two graphs are isomorphic or not would be greatly appreciated.

n	1	2	3	4	5	6	7	8	9
graphs	1	2	8	64	1024	32 768	2 097 152	268 435 456	$2^{36} > 6 \cdot 10^{10}$
nonisomorphic	1	2	4	11	34	156	1044	12 346	274 668

Other representations

Plane figures catch graphs for our eyes, but if a problem on graphs is to be *programmed*, then these figures are, to say the least, unsuitable. Integer matrices are ideal for computers, since every respectable programming language has array structures for these, and computers are good in crunching numbers.

Let $V_G = \{v_1, \ldots, v_n\}$ be ordered. The **adjacency matrix** of G is the $n \times n$-matrix M with entries $M_{ij} = 1$ or $M_{ij} = 0$ according to whether $v_iv_j \in G$ or $v_iv_j \notin G$. For instance, the graph in Example 1.1 has an adjacency matrix on the right. Notice that the adjacency matrix is always symmetric (with respect to its diagonal consisting of zeros).

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

A graph has usually many different adjacency matrices, one for each ordering of its set V_G of vertices. The following result is obvious from the definitions.

Theorem 1.1. *Two graphs G and H are isomorphic if and only if they have a common adjacency matrix. Moreover, two isomorphic graphs have exactly the same set of adjacency matrices.*

Graphs can also be represented by sets. For this, let $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ be a family of subsets of a set X, and define the **intersection graph** $G_{\mathcal{X}}$ as the graph with vertices $X_1, \ldots, X_n$, and edges X_iX_j for all i and j ($i \neq j$) with $X_i \cap X_j \neq \emptyset$.

Theorem 1.2. *Every graph is an intersection graph of some family of subsets.*

Proof. Let G be a graph, and define, for all $v \in G$, a set

$$X_v = \{\{v, u\} \mid vu \in G\}.$$

Then $X_u \cap X_v \neq \emptyset$ if and only if $uv \in G$. □

Let $s(G)$ be the smallest size of a base set X such that G can be represented as an intersection graph of a family of subsets of X, that is,

$$s(G) = \min\{|X| \mid G \cong G_{\mathcal{X}} \text{ for some } \mathcal{X} \subseteq 2^X\}.$$

How small can $s(G)$ be compared to the order ν_G (or the size ε_G) of the graph? It was shown by Kou, Stockmeyer and Wong (1976) that it is algorithmically difficult to determine the number $s(G)$ – the problem is NP-complete.

Example 1.2. As yet another example, let $A \subseteq \mathbb{N}$ be a finite set of natural numbers, and let $G_A = (A, E)$ be the graph with $rs \in E$ if and only if r and s (for $r \neq s$) have a common divisor > 1. As an exercise, we state: *All graphs can be represented in the form G_A for some set A of natural numbers.*

1.2 Subgraphs

Ideally, given a nice problem the local properties of a graph determine a solution. In these situations we deal with (small) parts of the graph (subgraphs), and a solution can be found to the problem by combining the information determined by the parts. For instance, as we shall later see, the existence of an Euler tour is very local, it depends only on the number of the neighbours of the vertices.

Degrees of vertices

DEFINITION. Let $v \in G$ be a vertex a graph G. The **neighbourhood** of v is the set

$$N_G(v) = \{u \in G \mid vu \in G\} .$$

The **degree** of v is the number of its neighbours:

$$d_G(v) = |N_G(v)| .$$

If $d_G(v) = 0$, then v is said to be **isolated** in G, and if $d_G(v) = 1$, then v is a **leaf** of the graph. The **minimum degree** and the **maximum degree** of G are defined as

$$\delta(G) = \min\{d_G(v) \mid v \in G\} \quad \text{and} \quad \Delta(G) = \max\{d_G(v) \mid v \in G\} .$$

The following lemma, due to EULER (1736), tells that if several people shake hands, then the number of hands shaken is even.

Lemma 1.1 (Handshaking lemma). *For each graph G,*

$$\sum_{v \in G} d_G(v) = 2 \cdot \varepsilon_G .$$

Moreover, the number of vertices of odd degree is even.

Proof. Every edge $e \in E_G$ has two ends. The second claim follows immediately from the first one. □

Lemma 1.1 holds equally well for multigraphs, when $d_G(v)$ is defined as the number of edges that have v as an end, and when *each loop vv is counted twice.*

Note that the degrees of a graph G do not determine G. Indeed, there are graphs $G = (V, E_G)$ and $H = (V, E_H)$ on the same set of vertices that are *not* isomorphic, but for which $d_G(v) = d_H(v)$ for all $v \in V$.

Subgraphs

DEFINITION. A graph H is a **subgraph** of a graph G, denoted by $H \subseteq G$, if $V_H \subseteq V_G$ and $E_H \subseteq E_G$. A subgraph $H \subseteq G$ **spans** G (and H is a **spanning subgraph** of G), if every vertex of G is in H, i.e., $V_H = V_G$.

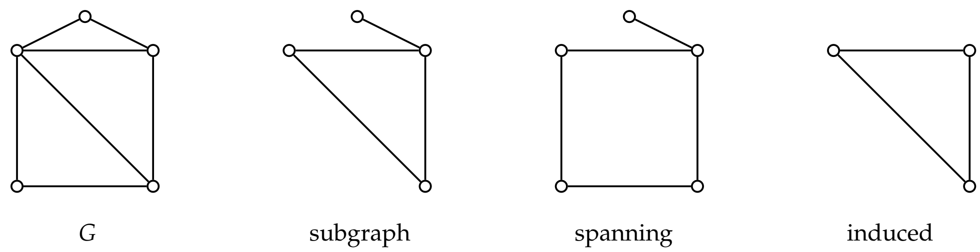
Also, a subgraph $H \subseteq G$ is an **induced subgraph**, if $E_H = E_G \cap E(V_H)$. In this case, H is **induced** by its set V_H of vertices.

In an induced subgraph $H \subseteq G$, the set E_H of edges consists of all $e \in E_G$ such that $e \in E(V_H)$. To each nonempty subset $A \subseteq V_G$, there corresponds a unique induced subgraph

$$G[A] = (A, E_G \cap E(A))\,.$$

To each subset $F \subseteq E_G$ of edges there corresponds a unique spanning subgraph of G,

$$G[F] = (V_G, F)\,.$$



For a set $F \subseteq E_G$ of edges, let

$$G{-}F = G[E_G \setminus F]$$

be the subgraph of G obtained by removing (only) the edges $e \in F$ from G. In particular, $G{-}e$ is obtained from G by removing $e \in G$.

Similarly, we write $G + F$, if each $e \in F$ (for $F \subseteq E(V_G)$) is added to G.

For a subset $A \subseteq V_G$ of vertices, we let $G{-}A \subseteq G$ be the subgraph induced by $V_G \setminus A$, that is,

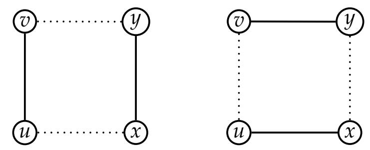
$$G{-}A = G[V_G \setminus A]\,,$$

and, *e.g.*, $G{-}v$ is obtained from G by removing the vertex v together with the edges that have v as their end.

Reconstruction Problem. The famous open problem, **Kelly-Ulam problem** or the **Reconstruction Conjecture**, states that *a graph of order at least* 3 *is determined up to isomorphism by its vertex deleted subgraphs* $G{-}v$ *(*$v \in G$*)*: if there exists a bijection $\alpha\colon V_G \to V_H$ such that $G{-}v \cong H{-}\alpha(v)$ for all v, then $G \cong H$.

2-switches

DEFINITION. For a graph G, a **2-switch** with respect to the edges $uv, xy \in G$ with $ux, vy \notin G$ replaces the edges uv and xy by ux and vy. Denote



$$G \xrightarrow{2s} H$$

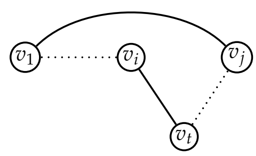
if there exists a finite sequence of 2-switches that carries G to H.

Note that if $G \xrightarrow{2s} H$ then also $H \xrightarrow{2s} G$ since we can apply the sequence of 2-switches in reverse order.

Before proving Berge's switching theorem we need the following tool.

Lemma 1.2. *Let G be a graph of order n with a degree sequence $d_1 \geq d_2 \geq \cdots \geq d_n$, where $d_G(v_i) = d_i$. Then there is a graph G' such that $G \xrightarrow{2s} G'$ with $N_{G'}(v_1) = \{v_2, \ldots, v_{d_1+1}\}$.*

Proof. Let $d = \Delta(G)$ $(= d_1)$. Suppose that there is a vertex v_i with $2 \leq i \leq d+1$ such that $v_1v_i \notin G$. Since $d_G(v_1) = d$, there exists a v_j with $j \geq d+2$ such that $v_1v_j \in G$. Here $d_i \geq d_j$, since $j > i$. Since $v_1v_j \in G$, there exists a v_t $(2 \leq t \leq n)$ such that $v_iv_t \in G$, but $v_jv_t \notin G$. We can now perform a 2-switch with respect to the vertices v_1, v_j, v_i, v_t. This gives a new graph H, where $v_1v_i \in H$ and $v_1v_j \notin H$, and the other neighbours of v_1 remain to be its neighbours.



When we repeat this process for all indices i with $v_1v_i \notin G$ for $2 \leq i \leq d+1$, we obtain a graph G' as required. □

Theorem 1.3 (BERGE (1973)). *Two graphs G and H on a common vertex set V satisfy $d_G(v) = d_H(v)$ for all $v \in V$ if and only if H can be obtained from G by a sequence of 2-switches.*

Proof. If $G \xrightarrow{2s} H$, then clearly H has the same degrees as G.

In converse, we use induction on the order ν_G. Let G and H have the same degrees. By Lemma 1.2, we have a vertex v and graphs G' and H' such that $G \xrightarrow{2s} G'$ and $H \xrightarrow{2s} H'$ with $N_{G'}(v) = N_{H'}(v)$. Now the graphs $G'-v$ and $H'-v$ have the same degrees. By the induction hypothesis, $G'-v \xrightarrow{2s} H'-v$, and thus also $G' \xrightarrow{2s} H'$. Finally, we observe that $H' \xrightarrow{2s} H$ by the 'reverse 2-switches', and this proves the claim. □

DEFINITION. Let $d_1, d_2, \ldots, d_n$ be a descending sequence of nonnegative integers, that is, $d_1 \geq d_2 \geq \cdots \geq d_n$. Such a sequence is said to be **graphical**, if there exists a graph $G = (V, E)$ with $V = \{v_1, v_2, \ldots, v_n\}$ such that $d_i = d_G(v_i)$ for all i.

Using the next result recursively one can decide whether a sequence of integers is graphical or not.

Theorem 1.4 (Havel **(1955),** Hakimi **(1962)).** *A sequence $d_1, d_2, \dots, d_n$ (with $d_1 \geq 1$ and $n \geq 2$) is graphical if and only if*

$$d_2 - 1,\ d_3 - 1, \dots, d_{d_1+1} - 1,\ d_{d_1+2},\ d_{d_1+3}, \dots, d_n \tag{1.1}$$

is graphical (when put into nonincreasing order).

Proof. ($\Leftarrow$) Consider G of order $n-1$ with vertices (and degrees)

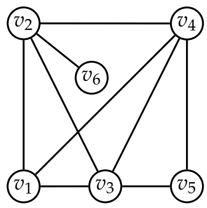
$$d_G(v_2) = d_2 - 1, \dots, d_G(v_{d_1+1}) = d_{d_1+1} - 1,$$
$$d_G(v_{d_1+2}) = d_{d_1+2}, \dots, d_G(v_n) = d_n$$

as in (1.1). Add a new vertex v_1 and the edges $v_1 v_i$ for all $i \in [2, d_{d_1+1}]$. Then in the new graph H, $d_H(v_1) = d_1$, and $d_H(v_i) = d_i$ for all i.

($\Rightarrow$) Assume $d_G(v_i) = d_i$. By Lemma 1.2 and Theorem 1.3, we can suppose that $N_G(v_1) = \{v_2, \dots, v_{d_1+1}\}$. But now the degree sequence of $G - v_1$ is in (1.1). □

Example 1.3. Consider the sequence $s = 4, 4, 4, 3, 2, 1$. By Theorem 1.4,

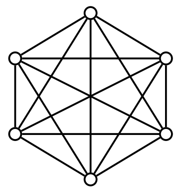
$$\begin{aligned} s \text{ is graphical} \iff & 3,3,2,1,1 \text{ is graphical} \\ & 2,1,1,0 \text{ is graphical} \\ & 0,0,0 \text{ is graphical.} \end{aligned}$$



The last sequence corresponds to a graph with no edges, and hence also our original sequence s is graphical. Indeed, the graph G on the right has this degree sequence.

Special graphs

Definition. A graph $G = (V, E)$ is **trivial**, if it has only one vertex, *i.e.*, $\nu_G = 1$; otherwise G is **nontrivial**.



The graph $G = K_V$ is the **complete graph** on V, if every two vertices are adjacent: $E = E(V)$. All complete graphs of order n are isomorphic with each other, and they will be denoted by K_n.

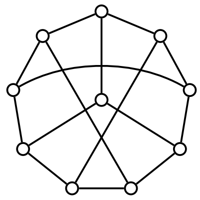
The **complement** of G is the graph $\overline{G}$ on V_G, where $E_{\overline{G}} = \{e \in E(V) \mid e \notin E_G\}$. The complements $G = \overline{K}_V$ of the complete graphs are called **discrete graphs**. In a discrete graph $E_G = \emptyset$. Clearly, all discrete graphs of order n are isomorphic with each other.

A graph G is said to be **regular**, if every vertex of G has the same degree. If this degree is equal to r, then G is r-**regular** or **regular of degree** r.

A discrete graph is 0-regular, and a complete graph K_n is $(n-1)$-regular. In particular, $\varepsilon_{K_n} = n(n-1)/2$, and therefore $\varepsilon_G \le n(n-1)/2$ for all graphs G that have order n.

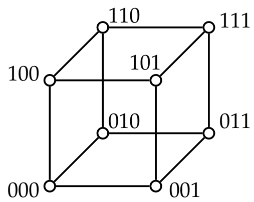
Many problems concerning (induced) subgraphs are algorithmically difficult. For instance, to find a maximal complete subgraph (a subgraph K_m of maximum order) of a graph is unlikely to be even in NP.

Example 1.4. The graph on the right is the **Petersen graph** that we will meet several times (drawn differently). It is a 3-regular graph of order 10.



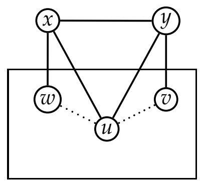
Example 1.5. Let $k \ge 1$ be an integer, and consider the set $\mathbb{B}^k$ of all binary strings of length k. For instance, $\mathbb{B}^3 = \{000, 001, 010, 100, 011, 101, 110, 111\}$. Let Q_k be the graph, called the k-**cube**, with $V_{Q_k} = \mathbb{B}^k$, where $uv \in Q_k$ if and only if the strings u and v differ in exactly one place.

The order of Q_k is $\nu_{Q_k} = 2^k$, the number of binary strings of length k. Also, Q_k is k-regular, and so, by the handshaking lemma, $\varepsilon_{Q_k} = k \cdot 2^{k-1}$. On the right we have the 3-cube, or simply the cube.



Example 1.6. Let $n \ge 4$ be any even number. We show by induction that there exists a 3-regular graph G with $\nu_G = n$. Notice that all 3-regular graphs have even order by the handshaking lemma.

If $n = 4$, then K_4 is 3-regular. Let G be a 3-regular graph of order $2m-2$, and suppose that $uv, uw \in E_G$. Let $V_H = V_G \cup \{x, y\}$, and $E_H = (E_G \setminus \{uv, uw\}) \cup \{ux, xv, uy, yw, xy\}$. Then H is 3-regular of order $2m$.



1.3 Paths and cycles

The most fundamental notions in graph theory are practically oriented. Indeed, many graph theoretical questions ask for optimal solutions to problems such as: find a shortest path (in a complex network) from a given point to another. This kind of problems can be difficult, or at least nontrivial, because there are usually choices what branch to choose when leaving an intermediate point.

Walks

DEFINITION. Let $e_i = u_i u_{i+1} \in G$ be edges of G for $i \in [1, k]$. The sequence $W = e_1 e_2 \dots e_k$ is a **walk of length** k **from** u_1 **to** u_{k+1}. Here e_i and e_{i+1} are compatible in the sense that e_i is adjacent to e_{i+1} for all $i \in [1, k-1]$.

We write, more informally,

$$W\colon u_1 \to u_2 \to \dots \to u_k \to u_{k+1} \quad \text{or} \quad W\colon u_1 \xrightarrow{k} u_{k+1}\,.$$

Write $u \xrightarrow{\star} v$ to say that there is a walk of some length from u to v. Here we understand that $W\colon u \xrightarrow{\star} v$ *is always a specific walk*, $W = e_1e_2 \dots e_k$, although we sometimes do not care to mention the edges e_i on it. The length of a walk W is denoted by $|W|$.

DEFINITION. Let $W = e_1e_2 \dots e_k$ ($e_i = u_iu_{i+1}$) be a walk.
W is **closed**, if $u_1 = u_{k+1}$.
W is a **path**, if $u_i \neq u_j$ for all $i \neq j$.
W is a **cycle**, if it is closed, and $u_i \neq u_j$ for $i \neq j$ except that $u_1 = u_{k+1}$.
W is a **trivial path**, if its length is 0. A trivial path has no edges.
For a walk $W\colon u = u_1 \to \dots \to u_{k+1} = v$, also

$$W^{-1}\colon v = u_{k+1} \to \dots \to u_1 = u$$

is a walk in G, called the **inverse walk** of W.

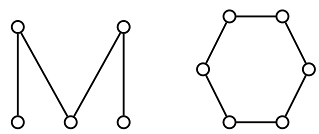
A vertex u is an **end** of a path P, if P starts or ends in u.

The **join** of two walks $W_1\colon u \xrightarrow{\star} v$ and $W_2\colon v \xrightarrow{\star} w$ is the walk $W_1W_2\colon u \xrightarrow{\star} w$. (Here the end v must be common to the walks.)

Paths P and Q are **disjoint**, if they have no vertices in common, and they are **independent**, if they can share only their ends.

Clearly, the inverse walk P^{-1} of a path P is a path (the **inverse path** of P). The join of two paths need not be a path.

A (sub)graph, which is a path (cycle) of length $k-1$ (k, resp.) having k vertices is denoted by P_k (C_k, resp.). If k is even (odd), we say that the path or cycle is **even** (**odd**). Clearly, all paths of length k are isomorphic. The same holds for cycles of fixed length.



P_5 C_6

Lemma 1.3. *Each walk $W\colon u \xrightarrow{\star} v$ with $u \neq v$ contains a path $P\colon u \xrightarrow{\star} v$, that is, there is a path $P\colon u \xrightarrow{\star} v$ that is obtained from W by removing edges and vertices.*

Proof. Let $W\colon u = u_1 \to \dots \to u_{k+1} = v$. Let $i < j$ be indices such that $u_i = u_j$. If no such i and j exist, then W, itself, is a path. Otherwise, in $W = W_1W_2W_3\colon u \xrightarrow{\star} u_i \xrightarrow{\star} u_j \xrightarrow{\star} v$ the portion $U_1 = W_1W_3\colon u \xrightarrow{\star} u_i = u_j \xrightarrow{\star} v$ is a shorter walk. By repeating this argument, we obtain a sequence $U_1, U_2, \dots, U_m$ of walks $u \xrightarrow{\star} v$ with $|W| > |U_1| > \dots > |U_m|$. When the procedure stops, we have a path as required. (Notice that in the above it may very well be that W_1 or W_3 is a trivial walk.) □

DEFINITION. If there exists a walk (and hence a path) from u to v in G, let

$$d_G(u,v) = \min\{k \mid u \xrightarrow{k} v\}$$

be the **distance** between u and v. If there are no walks $u \xrightarrow{\star} v$, let $d_G(u,v) = \infty$ by convention. A graph G is **connected**, if $d_G(u,v) < \infty$ for all $u,v \in G$; otherwise, it is **disconnected**. The maximal connected subgraphs of G are its **connected components**. Denote

$$c(G) = \text{ the number of connected components of } G\,.$$

If $c(G) = 1$, then G is, of course, connected.

The maximality condition means that a subgraph $H \subseteq G$ is a connected component if and only if H is connected and there are no edges leaving H, i.e., for every vertex $v \notin H$, the subgraph $G[V_H \cup \{v\}]$ is disconnected. Apparently, every connected component is an induced subgraph, and

$$N^*_G(v) = \{u \mid d_G(v,u) < \infty\}$$

is *the* connected component of G that contains $v \in G$. In particular, the connected components form a partition of G.

Shortest paths

DEFINITION. Let G^α be an edge weighted graph, that is, G^α is a graph G together with a weight function $\alpha\colon E_G \to \mathbb{R}$ on its edges. For $H \subseteq G$, let

$$\alpha(H) = \sum_{e \in H} \alpha(e)$$

be the (total) **weight** of H. In particular, if $P = e_1 e_2 \dots e_k$ is a path, then its weight is $\alpha(P) = \sum_{i=1}^{k} \alpha(e_i)$. The **minimum weighted distance** between two vertices is

$$d^\alpha_G(u,v) = \min\{\alpha(P) \mid P\colon u \xrightarrow{\star} v\}\,.$$

In extremal problems we seek for optimal subgraphs $H \subseteq G$ satisfying specific conditions. In practice we encounter situations where G might represent

- a distribution or transportation network (say, for mail), where the weights on edges are *distances*, travel *expenses*, or *rates of flow* in the network;
- a system of channels in (tele)communication or computer architecture, where the weights present the rate of *unreliability* or *frequency of action* of the connections;
- a model of chemical bonds, where the weights measure molecular *attraction*.

In these examples we look for a subgraph with the smallest weight, and which connects two given vertices, or all vertices (if we want to travel around). On the other hand, if the graph represents a network of pipelines, the weights are volumes or capacities, and then one wants to find a subgraph with the maximum weight.

We consider the minimum problem. For this, let G be a graph with an integer weight function $\alpha\colon E_G \to \mathbb{N}$. In this case, call $\alpha(uv)$ the **length** of uv.

The **shortest path problem**: *Given a connected graph G with a weight function $\alpha\colon E_G \to \mathbb{N}$, find $d_G^\alpha(u,v)$ for given $u, v \in G$.*

Assume that G is a connected graph. Dijkstra's algorithm solves the problem for every pair u, v, where u is a fixed starting point and $v \in G$. Let us make the convention that $\alpha(uv) = \infty$, if $uv \notin G$.

Dijkstra's algorithm:

(i) Set $u_0 = u$, $t(u_0) = 0$ and $t(v) = \infty$ for all $v \neq u_0$.

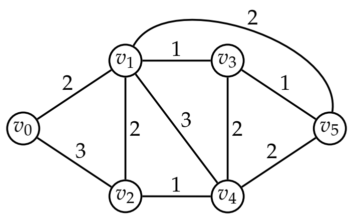
(ii) For $i \in [0, \nu_G - 1]$: for each $v \notin \{u_1, \dots, u_i\}$,

$$\text{replace } \; t(v) \; \text{ by } \; \min\{t(v), t(u_i) + \alpha(u_i v)\}\,.$$

Let $u_{i+1} \notin \{u_1, \dots, u_i\}$ be *any* vertex with the least value $t(u_{i+1})$.

(iii) Conclusion: $d_G^\alpha(u,v) = t(v)$.

Example 1.7. Consider the following weighted graph G. Apply Dijkstra's algorithm to the vertex v_0.



- $u_0 = v_0$, $t(u_0) = 0$, others are ∞.
- $t(v_1) = \min\{\infty, 2\} = 2$, $t(v_2) = \min\{\infty, 3\} = 3$, others are ∞. Thus $u_1 = v_1$.
- $t(v_2) = \min\{3, t(u_1) + \alpha(u_1 v_2)\} = \min\{3, 4\} = 3$, $t(v_3) = 2 + 1 = 3$, $t(v_4) = 2 + 3 = 5$, $t(v_5) = 2 + 2 = 4$. Thus choose $u_2 = v_3$.
- $t(v_2) = \min\{3, \infty\} = 3$, $t(v_4) = \min\{5, 3 + 2\} = 5$, $t(v_5) = \min\{4, 3 + 1\} = 4$. Thus set $u_3 = v_2$.
- $t(v_4) = \min\{5, 3 + 1\} = 4$, $t(v_5) = \min\{4, \infty\} = 4$. Thus choose $u_4 = v_4$.
- $t(v_5) = \min\{4, 4 + 1\} = 4$. The algorithm stops.

We have obtained:

$$t(v_1) = 2,\; t(v_2) = 3,\; t(v_3) = 3,\; t(v_4) = 4,\; t(v_5) = 4\,.$$

These are the minimal weights from v_0 to each v_i.

The steps of the algorithm can also be rewritten as a table:

v_1	**2**	-	-	-	-
v_2	3	3	**3**	-	-
v_3	∞	**3**	-	-	-
v_4	∞	5	5	**4**	-
v_5	∞	4	4	4	**4**

The correctness of Dijkstra's algorithm can verified be as follows.

Let $v \in V$ be any vertex, and let $P\colon u_0 \xrightarrow{\star} u \xrightarrow{\star} v$ be a shortest path from u_0 to v, where u is any vertex $u \neq v$ on such a path, possibly $u = u_0$. Then, clearly, the first part of the path, $u_0 \xrightarrow{\star} u$, is a shortest path from u_0 to u, and the latter part $u \xrightarrow{\star} v$ is a shortest path from u to v. Therefore, the length of the path P equals the sum of the weights of $u_0 \xrightarrow{\star} u$ and $u \xrightarrow{\star} v$. Dijkstra's algorithm makes use of this observation iteratively.

2

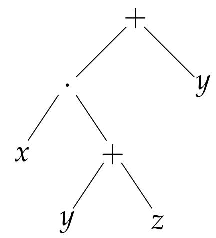
Connectivity of Graphs

2.1 Bipartite graphs and trees

In problems such as the shortest path problem we look for minimum solutions that satisfy the given requirements. The solutions in these cases are usually subgraphs without cycles. Such connected graphs will be called trees, and they are used, *e.g.*, in search algorithms for databases. For concrete applications in this respect, see

T.H. CORMEN, C.E. LEISERSON AND R.L. RIVEST, "Introduction to Algorithms", MIT Press, 1993.

Certain structures with operations are representable as trees. These trees are sometimes called *construction trees*, *decomposition trees*, *factorization trees* or *grammatical trees*. Grammatical trees occur especially in linguistics, where syntactic structures of sentences are analyzed. On the right there is a tree of operations for the arithmetic formula $x \cdot (y + z) + y$.



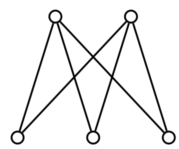
Bipartite graphs

DEFINITION. A graph G is called **bipartite**, if V_G has a partition to two subsets X and Y such that each edge $uv \in G$ connects a vertex of X and a vertex of Y. In this case, (X, Y) is a **bipartition** of G, and G is (X, Y)-**bipartite**.

A bipartite graph G (as in the above) is **complete** (m, k)-**bipartite**, if $|X| = m$, $|Y| = k$, and $uv \in G$ for all $u \in X$ and $v \in Y$.

All complete (m, k)-bipartite graphs are isomorphic. Let $K_{m,k}$ denote such a graph.

A subset $X \subseteq V_G$ is **stable**, if $G[X]$ is a discrete graph.



$K_{2,3}$

The following result is clear from the definitions.

Theorem 2.1. *A graph G is bipartite if and only if V_G has a partition to two stable subsets.*

Example 2.1. The k-cube Q_k of Example 1.5 is bipartite for all k. Indeed, consider $A = \{u \mid u \text{ has an even number of } 1's\}$ and $B = \{u \mid u \text{ has an odd number of } 1's\}$. Clearly, these sets partition $\mathbb{B}^k$, and they are stable in Q_k.

Theorem 2.2. *A graph G is bipartite if and only if G it has no odd cycles (as subgraph).*

Proof. ($\Rightarrow$) Observe that if G is (X,Y)-bipartite, then so are all its subgraphs. However, an odd cycle C_{2k+1} is not bipartite.

($\Leftarrow$) Suppose that all cycles in G are even. First, we note that it suffices to show the claim for connected graphs. Indeed, if G is disconnected, then each cycle of G is contained in one of the connected components $G_1,\dots,G_p$ of G. If G_i is (X_i,Y_i)-bipartite, then G has the bipartition $(X_1\cup X_2\cup\dots\cup X_p, Y_1\cup Y_2\cup\dots\cup Y_p)$.

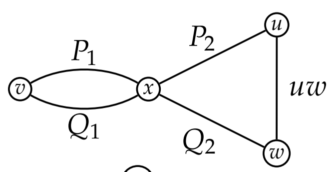
Assume thus that G is connected. Let $v\in G$ be a chosen vertex, and define

$$X=\{x \mid d_G(v,x)\ \text{ is even}\}\quad\text{ and }\quad Y=\{y\mid d_G(v,y)\ \text{ is odd}\}\,.$$

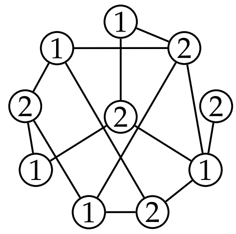
Since G is connected, $V_G = X\cup Y$. Also, by the definition of distance, $X\cap Y=\emptyset$.

Let then $u,w\in G$ be both in X or both in Y, and let $P\colon v\xrightarrow{\star} u$ and $Q\colon v\xrightarrow{\star} w$ be (among the) shortest paths from v to u and w. Assume that x is the last common vertex of P and Q: $P=P_1P_2$, $Q=Q_1Q_2$, where $P_2\colon x\xrightarrow{\star} u$ and $Q_2\colon x\xrightarrow{\star} w$ are independent. Since P and Q are shortest paths, P_1 and Q_1 are shortest paths $v\xrightarrow{\star} x$. Consequently, $|P_1|=|Q_1|$.

Thus $|P_2|$ and $|Q_2|$ have the same parity and hence the sum $|P_2|+|Q_2|$ is even, i.e., the path $P_2^{-1}Q_2$ is even, and so $uw\notin E_G$ by assumption. Therefore X and Y are stable subsets, and G is bipartite as claimed. □



Checking whether a graph is bipartite is easy. Indeed, this can be done by using two 'opposite' colours, say 1 and 2. Start from any vertex v_1, and colour it by 1. Then colour the neighbours of v_1 by 2, and proceed by colouring all neighbours of an already coloured vertex by the opposite colour.



If the whole graph can be coloured without contradiction, then G is (X,Y)-bipartite, where X consists of those vertices with colour 1, and Y of those vertices with colour 2; otherwise, at some point one of the vertices gets both colours, and in this case, G is not bipartite.

Example 2.2 (ERDÖS **(1965)).** We show that each graph G has a bipartite subgraph $H\subseteq G$ such that $\varepsilon_H\ge\frac{1}{2}\varepsilon_G$. Indeed, let $V_G = X\cup Y$ be a partition such that the number of edges between X and Y is maximum. Denote

$$F=E_G\cap\{uv\mid u\in X, v\in Y\}\,,$$

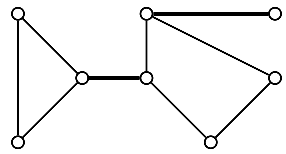
and let $H=G[F]$. Obviously H is a spanning subgraph, and it is bipartite.

By the maximum condition, $d_H(v)\ge d_G(v)/2$, since, otherwise, v is on the wrong side. (That is, if $v\in X$, then the pair $X'=X\setminus\{v\}$, $Y'=Y\cup\{v\}$ does better that the pair X,Y.) Now

$$\varepsilon_H=\frac{1}{2}\sum_{v\in H} d_H(v)\ge\frac{1}{2}\sum_{v\in G}\frac{1}{2}d_G(v)=\frac{1}{2}\varepsilon_G\,.$$

Bridges

DEFINITION. An edge $e \in G$ is a **bridge** of the graph G, if $G-e$ has more connected components than G, that is, if $c(G-e) > c(G)$. In particular, and most importantly, an edge e in a connected G is a bridge if and only if $G-e$ is disconnected.



On the right (only) the two horizontal lines are bridges.

We note that, for each edge $e \in G$,

$$e = uv \text{ is a bridge } \iff u, v \text{ in different connected components of } G-e\,.$$

Theorem 2.3. *An edge $e \in G$ is a bridge if and only if e is not in any cycle of G.*

Proof. ($\Rightarrow$) If there is a cycle in G containing e, say $C = PeQ$, then $QP\colon v \xrightarrow{\star} u$ is a path in $G-e$, and so e is not a bridge.

($\Leftarrow$) If $e = uv$ is not a bridge, then u and v are in the same connected component of $G-e$, and there is a path $P\colon v \xrightarrow{\star} u$ in $G-e$. Now, $eP\colon u \to v \xrightarrow{\star} u$ is a cycle in G containing e. □

Lemma 2.1. *Let e be a bridge in a connected graph G.*

(i) Then $c(G-e) = 2$.

(ii) Let H be a connected component of $G-e$. If $f \in H$ is a bridge of H, then f is a bridge of G.

Proof. For (i), let $e = uv$. Since e is a bridge, the ends u and v are not connected in $G-e$. Let $w \in G$. Since G is connected, there exists a path $P\colon w \xrightarrow{\star} v$ in G. This is a path of $G-e$, unless $P\colon w \xrightarrow{\star} u \to v$ contains $e = uv$, in which case the part $w \xrightarrow{\star} u$ is a path in $G-e$.

For (ii), if $f \in H$ belongs to a cycle C of G, then C does not contain e (since e is in no cycle), and therefore C is inside H, and f is not a bridge of H. □

Trees

DEFINITION. A graph is called **acyclic**, if it has no cycles. An acyclic graph is also called a **forest**. A **tree** is a connected acyclic graph.

By Theorem 2.3 and the definition of a tree, we have

Corollary 2.1. *A connected graph is a tree if and only if all its edges are bridges.*

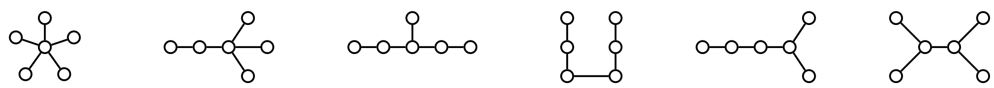
Example 2.3. The following enumeration result for trees has many different proofs, the first of which was given by CAYLEY in 1889: *There are n^{n-2} trees on a vertex set V of n elements.* We omit the proof.

On the other hand, there are only a few trees *up to isomorphism*:

n	1	2	3	4	5	6	7	8
trees	1	1	1	2	3	6	11	23

n	9	10	11	12	13	14	15	16
trees	47	106	235	551	1301	3159	7741	19 320

The nonisomorphic trees of order 6 are:



We say that a path $P\colon u \xrightarrow{\star} v$ is **maximal** in a graph G, if there are no edges $e \in G$ for which Pe or eP is a path. Such paths exist, because ν_G is finite.

Lemma 2.2. *Let $P\colon u \xrightarrow{\star} v$ be a maximal path in a graph G. Then $N_G(v) \subseteq P$. Moreover, if G is acyclic, then $d_G(v) = 1$.*

Proof. If $e = vw \in E_G$ with $w \notin P$, then also Pe is a path, which contradicts the maximality assumption for P. Hence $N_G(v) \subseteq P$. For acyclic graphs, if $wv \in G$, then w belongs to P, and wv is necessarily the last edge of P in order to avoid cycles. □

Corollary 2.2. *Each tree T with $\nu_T \geq 2$ has at least two leaves.*

Proof. Since T is acyclic, both ends of a maximal path have degree one. □

Theorem 2.4. *The following are equivalent for a graph T.*

(i) *T is a tree.*
(ii) *Any two vertices are connected in T by a unique path.*
(iii) *T is acyclic and $\varepsilon_T = \nu_T - 1$.*

Proof. Let $\nu_T = n$. If $n = 1$, then the claim is trivial. Suppose thus that $n \geq 2$.

(i)⇒(ii) Let T be a tree. Assume the claim does not hold, and let $P, Q\colon u \xrightarrow{\star} v$ be two different paths between the same vertices u and v. Suppose that $|P| \geq |Q|$. Since $P \neq Q$, there exists an edge e which belongs to P but not to Q. Each edge of T is a bridge, and therefore u and v belong to different connected components of $T{-}e$. Hence e must also belong to Q; a contradiction.

(ii)⇒(iii) We prove the claim by induction on n. Clearly, the claim holds for $n = 2$, and suppose it holds for graphs of order less than n. Let T be any graph of order n satisfying (ii). In particular, T is connected, and it is clearly acyclic.

Let $P\colon u \xrightarrow{\star} v$ be a maximal path in T. By Lemma 2.2, we have $d_T(v) = 1$. In this case, $P\colon u \xrightarrow{\star} w \to v$, where vw is the unique edge having an end v. The subgraph $T-v$ is connected, and it satisfies the condition (ii). By induction hypothesis, $\varepsilon_{T-v} = n-2$, and so $\varepsilon_T = \varepsilon_{T-v} + 1 = n-1$, and the claim follows.

(iii)⇒(i) Assume (iii) holds for T. We need to show that T is connected. Indeed, let the connected components of T be $T_i = (V_i, E_i)$, for $i \in [1,k]$. Since T is acyclic, so are the connected graphs T_i, and hence they are trees, for which we have proved that $|E_i| = |V_i| - 1$. Now, $\nu_T = \sum_{i=1}^k |V_i|$, and $\varepsilon_T = \sum_{i=1}^k |E_i|$. Therefore,

$$n - 1 = \varepsilon_T = \sum_{i=1}^{k}(|V_i| - 1) = \sum_{i=1}^{k} |V_i| - k = n - k\,,$$

which gives that $k = 1$, that is, T is connected. □

Example 2.4. Consider a cup tournament of n teams. If during a round there are k teams left in the tournament, then these are divided into $\lfloor k \rfloor$ pairs, and from each pair only the winner continues. If k is odd, then one of the teams goes to the next round without having to play. How many plays are needed to determine the winner?

So if there are 14 teams, after the first round 7 teams continue, and after the second round 4 teams continue, then 2. So 13 plays are needed in this example.

The answer to our problem is $n-1$, since the cup tournament is a tree, where a play corresponds to an edge of the tree.

Spanning trees

Theorem 2.5. *Each connected graph has a* **spanning tree**, *that is, a spanning graph that is a tree.*

Proof. Let $T \subseteq G$ be a maximum order subtree of G (i.e., subgraph that is a tree). If $V_T \neq V_G$, there exists an edge $uv \notin E_G$ such that $u \in T$ and $v \notin T$. But then T is not maximal; a contradiction. □

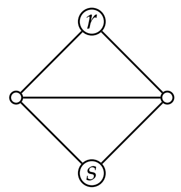
Corollary 2.3. *For each connected graph G, $\varepsilon_G \geq \nu_G - 1$. Moreover, a connected graph G is a tree if and only if $\varepsilon_G = \nu_G - 1$.*

Proof. Let T be a spanning tree of G. Then $\varepsilon_G \geq \varepsilon_T = \nu_T - 1 = \nu_G - 1$. The second claim is also clear. □

Example 2.5. In **Shannon's switching game** a positive player P and a negative player N play on a graph G with two special vertices: a **source** s and a **sink** r. P and N alternate turns so that P designates an edge by $+$, and N by $-$. Each edge can be designated at most once. It is P's purpose to designate a path $s \xrightarrow{\star} r$ (that is, to designate all edges in one such path), and N tries to block all paths $s \xrightarrow{\star} r$ (that is, to designate at least one edge in each such path). We say that a game (G, s, r) is

- **positive**, if P has a winning strategy no matter who begins the game,
- **negative**, if N has a winning strategy no matter who begins the game,
- **neutral**, if the winner depends on who begins the game.

The game on the right is neutral.



LEHMAN proved in 1964 that *Shannon's switching game* (G, s, r) *is positive if and only if there exists* $H \subseteq G$ *such that* H *contains* s *and* r *and* H *has two spanning trees with no edges in common.*

In the other direction the claim can be proved along the following lines. Assume that there exists a subgraph H containing s and r and that has two spanning trees with no edges in common. Then P plays as follows. If N marks by $-$ an edge from one of the two trees, then P marks by $+$ an edge in the other tree such that this edge reconnects the broken tree. In this way, P always has two spanning trees for the subgraph H with only edges marked by $+$ in common.

In converse the claim is considerably more difficult to prove.

There remains the problem to characterize those Shannon's switching games (G, s, r) that are neutral (negative, respectively).

The connector problem

To build a network connecting n nodes (towns, computers, chips in a computer) it is desirable to decrease the cost of construction of the links to the minimum. This is the **connector problem**. In graph theoretical terms we wish to find an **optimal spanning subgraph** of a weighted graph. Such an optimal subgraph is clearly a spanning tree, for, otherwise a deletion of any nonbridge will reduce the total weight of the subgraph.

Let then G^α be a graph G together with a weight function $\alpha\colon E_G \to \mathbb{R}^+$ (positive reals) on the edges. Kruskal's algorithm (also known as the **greedy algorithm**) provides a solution to the connector problem.

Kruskal's algorithm: For a connected and weighted graph G^α of order n:

(i) Let e_1 be an edge of smallest weight, and set $E_1 = \{e_1\}$.

(ii) For each $i = 2, 3, \dots, n-1$ in this order, choose an edge $e_i \notin E_{i-1}$ of smallest possible weight such that e_i does not produce a cycle when added to $G[E_{i-1}]$, and let $E_i = E_{i-1} \cup \{e_i\}$.

The final outcome is $T = (V_G, E_{n-1})$.

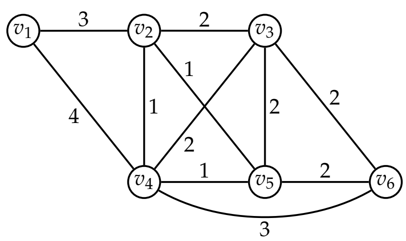
By the construction, $T = (V_G, E_{n-1})$ is a spanning tree of G, because it contains no cycles, it is connected and has $n-1$ edges. We now show that T has the minimum total weight among the spanning trees of G.

Suppose T_1 is any spanning tree of G. Let e_k be the first edge produced by the algorithm that is not in T_1. If we add e_k to T_1, then a cycle C containing e_k is created. Also, C must contain an edge e that is not in T. When we replace e by e_k in T_1, we still have a spanning tree, say T_2. However, by the construction, $\alpha(e_k) \leq \alpha(e)$, and therefore $\alpha(T_2) \leq \alpha(T_1)$. Note that T_2 has more edges in common with T than T_1.

Repeating the above procedure, we can transform T_1 to T by replacing edges, one by one, such that the total weight does not increase. We deduce that $\alpha(T) \leq \alpha(T_1)$.

The outcome of Kruskal's algorithm need not be unique. Indeed, there may exist several optimal spanning trees (with the same weight, of course) for a graph.

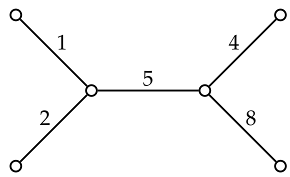
Example 2.6. When applied to the weighted graph on the right, the algorithm produces the sequence: $e_1 = v_2v_4$, $e_2 = v_4v_5$, $e_3 = v_3v_6$, $e_4 = v_2v_3$ and $e_5 = v_1v_2$. The total weight of the spanning tree is thus 9.
Also, the selection $e_1 = v_2v_5$, $e_2 = v_4v_5$, $e_3 = v_5v_6$, $e_4 = v_3v_6$, $e_5 = v_1v_2$ gives another optimal solution (of weight 9).



Problem. Consider trees T with weight functions $\alpha\colon E_T \to \mathbb{N}$. Each tree T of order n has exactly $\binom{n}{2}$ paths. (Why is this so?) *Does there exist a weighted tree T^α of order n such that the (total) weights of its paths are* $1, 2, \ldots, \binom{n}{2}$*?*

In such a weighted tree T^α different paths have different weights, and each $i \in [1, \binom{n}{2}]$ is a weight of one path. Also, α must be injective.

No solutions are known for any $n \geq 7$.

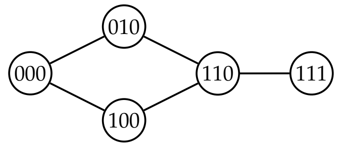


TAYLOR (1977) proved: *if T of order n exists, then necessarily $n = k^2$ or $n = k^2 + 2$ for some $k \geq 1$.*

Example 2.7. A computer network can be presented as a graph G, where the vertices are the node computers, and the edges indicate the direct links. Each computer v has an *address* $a(v)$, a bit string (of zeros and ones). The **length** of an address is the number of its bits. A message that is sent to v is preceded by the address $a(v)$. The **Hamming distance** $h(a(v), a(u))$ of two addresses of the same length is the number of places, where $a(v)$ and $a(u)$ differ; e.g., $h(00010, 01100) = 3$ and $h(10000, 00000) = 1$.

It would be a good way to address the vertices so that the Hamming distance of two vertices is the same as their distance in G. In particular, if two vertices were adjacent, their addresses should differ by one symbol. This would make it easier for a node computer to forward a message.

A graph G is said to be **addressable**, if it has an addressing a such that $d_G(u,v) = h(a(u), a(v))$.



We prove that *every tree T is addressable. Moreover, the addresses of the vertices of T can be chosen to be of length $\nu_T - 1$.*

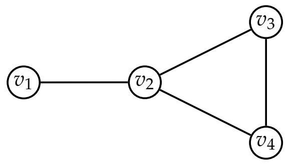
The proof goes by induction. If $\nu_T \leq 2$, then the claim is obvious. In the case $\nu_T = 2$, the addresses of the vertices are simply 0 and 1.

Let then $V_T = \{v_1, \dots, v_{k+1}\}$, and assume that $d_T(v_1) = 1$ (a leaf) and $v_1v_2 \in T$. By the induction hypothesis, we can address the tree $T - v_1$ by addresses of length $k-1$. We change this addressing: let a_i be the address of v_i in $T - v_1$, and change it to $0a_i$. Set the address of v_1 to $1a_2$. It is now easy to see that we have obtained an addressing for T as required.

The triangle K_3 is not addressable. In order to gain more generality, we modify the addressing for general graphs by introducing a special symbol $*$ in addition to 0 and 1. A **star address** will be a sequence of these three symbols. The Hamming distance remains as it was, that is, $h(u,v)$ is the number of places, where u and v have a different symbol 0 or 1. The special symbol $*$ does not affect $h(u,v)$. So, $h(10 * *01, 0 * *101) = 1$ and $h(1 * * * **, *00 * **) = 0$. We still want to have $h(u,v) = d_G(u,v)$.

We star address this graph as follows:

$$a(v_1) = 0000\,, \qquad a(v_2) = 10*0\,,$$
$$a(v_3) = 1*01\,, \qquad a(v_4) = **11\,.$$



These addresses have length 4. Can you design a star addressing with addresses of length 3?

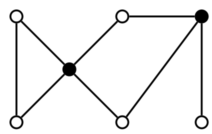
WINKLER proved in 1983 a rather unexpected result: *The minimum star address length of a graph G is at most $\nu_G - 1$.*

For the proof of this, see VAN LINT AND WILSON, "A Course in Combinatorics".

2.2 Connectivity

Spanning trees are often optimal solutions to problems, where cost is the criterion. We may also wish to construct graphs that are as simple as possible, but where two vertices are always connected by at least two independent paths. These problems occur especially in different aspects of fault tolerance and reliability of networks, where one has to make sure that a break-down of one connection does not affect the functionality of the network. Similarly, in a reliable network we require that a break-down of a node (computer) should not result in the inactivity of the whole network.

Separating sets



DEFINITION. A vertex $v \in G$ is a **cut vertex**, if $c(G-v) > c(G)$. A subset $S \subseteq V_G$ is a **separating set**, if $G-S$ is disconnected. We also say that S **separates** the vertices u and v and it is a (u,v)-**separating set**, if u and v belong to different connected components of $G-S$.

If G is connected, then v is a cut vertex if and only if $G-v$ is disconnected, that is, $\{v\}$ is a separating set. The following lemma is immediate.

Lemma 2.3. *If $S \subseteq V_G$ separates u and v, then every path $P\colon u \xrightarrow{\star} v$ visits a vertex of S.*

Lemma 2.4. *If a connected graph G has no separating sets, then it is a complete graph.*

Proof. If $\nu_G \leq 2$, then the claim is clear. For $\nu_G \geq 3$, assume that G is not complete, and let $uv \notin G$. Now $V_G \setminus \{u,v\}$ is a separating set. The claim follows from this. □

DEFINITION. The **(vertex) connectivity number** $\kappa(G)$ of G is defined as

$$\kappa(G) = \min\{k \mid k = |S|,\ G-S \text{ disconnected or trivial}, S \subseteq V_G\}\,.$$

A graph G is k-**connected**, if $\kappa(G) \geq k$.

In other words,

- $\kappa(G) = 0$, if G is disconnected,
- $\kappa(G) = \nu_G - 1$, if G is a complete graph, and
- otherwise $\kappa(G)$ equals the minimum size of a separating set of G.

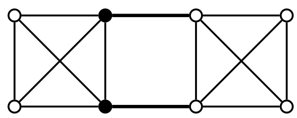
Clearly, if G is connected, then it is 1-connected.

DEFINITION. An **edge cut** F of G consists of edges so that $G-F$ is disconnected. Let

$$\kappa'(G) = \min\{k \mid k = |F|,\ G-F \text{ disconnected}, F \subseteq E_G\}\,.$$

For trivial graphs, let $\kappa'(G) = 0$. A graph G is k-**edge connected**, if $\kappa'(G) \geq k$. A minimal edge cut $F \subseteq E_G$ is a **bond** ($F \setminus \{e\}$ is not an edge cut for any $e \in F$).

Example 2.8. Again, if G is disconnected, then $\kappa'(G) = 0$. On the right, $\kappa(G) = 2$ and $\kappa'(G) = 2$. Notice that the minimum degree is $\delta(G) = 3$.



Lemma 2.5. *Let G be connected. If $e = uv$ is a bridge, then either $G = K_2$ or one of u or v is a cut vertex.*

Proof. Assume that $G \neq K_2$ and thus that $\nu_G \geq 3$, since G is connected. Let $G_u = N^*_{G-e}(u)$ and $G_v = N^*_{G-e}(v)$ be the connected components of $G{-}e$ containing u and v. Now, either $\nu_{G_u} \geq 2$ (and u is a cut vertex) or $\nu_{G_v} \geq 2$ (and v is a cut vertex). □

Lemma 2.6. *If F be a bond of a connected graph G, then $c(G{-}F) = 2$.*

Proof. Since $G{-}F$ is disconnected, and F is minimal, the subgraph $H = G{-}(F \setminus \{e\})$ is connected for given $e \in F$. Hence e is a bridge in H. By Lemma 2.1, $c(H{-}e) = 2$, and thus $c(G{-}F) = 2$, since $H{-}e = G{-}F$. □

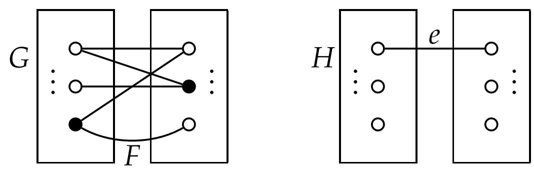
Theorem 2.6 (WHITNEY **(1932)).** *For any graph G,*

$$\kappa(G) \leq \kappa'(G) \leq \delta(G)\,.$$

Proof. Assume G is nontrivial. Clearly, $\kappa'(G) \leq \delta(G)$, since if we remove all edges with an end v, we disconnect G. If $\kappa'(G) = 0$, then G is disconnected, and in this case also $\kappa(G) = 0$. If $\kappa'(G) = 1$, then G is connected and contains a bridge. By Lemma 2.5, either $G = K_2$ or G has a cut vertex. In both of these cases, also $\kappa(G) = 1$.

Assume then that $\kappa'(G) \geq 2$. Let F be an edge cut of G with $|F| = \kappa'(G)$, and let $e = uv \in F$. Then F is a bond, and $G{-}F$ has two connected components.

Consider the connected subgraph $H = G{-}(F \setminus \{e\}) = (G{-}F) + e$, where e is a bridge.



Now for each $f \in F \setminus \{e\}$ choose an end different from u and v. (The choices for different edges need not be different.) Note that since $f \neq e$, either end of f is different from u or v. Let S be the collection of these choices. Thus $|S| \leq |F| - 1 = \kappa'(G) - 1$, and $G{-}S$ does not contain edges from $F \setminus \{e\}$.

If $G{-}S$ is disconnected, then S is a separating set and so $\kappa(G) \leq |S| \leq \kappa'(G) - 1$ and we are done. On the other hand, if $G{-}S$ is connected, then either $G{-}S = K_2$ $(= e)$, or either u or v (or both) is a cut vertex of $G{-}S$ (since $H{-}S = G{-}S$, and therefore $G{-}S \subseteq H$ is an induced subgraph of H). In both of these cases, there is a vertex of $G{-}S$, whose removal results in a trivial or a disconnected graph. In conclusion, $\kappa(G) \leq |S| + 1 \leq \kappa'(G)$, and the claim follows. □

Menger's theorem

Theorem 2.7 (MENGER **(1927)).** *Let $u, v \in G$ be nonadjacent vertices of a connected graph G. Then the minimum number of vertices separating u and v is equal to the maximum number of independent paths from u to v.*

Proof. If a subset $S \subseteq V_G$ is (u, v)-separating, then every path $u \xrightarrow{\star} v$ of G visits S. Hence $|S|$ is at least the number of independent paths from u to v.

Conversely, we use induction on $m = \nu_G + \varepsilon_G$ to show that if $S = \{w_1, w_2, \dots, w_k\}$ is a (u,v)-separating set of the smallest size, then G has at least (and thus exactly) k independent paths $u \xrightarrow{\star} v$.

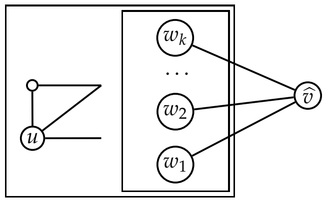
The case for $k = 1$ is clear, and this takes care of the small values of m, required for the induction.

(1) Assume first that u and v have a common neighbour $w \in N_G(u) \cap N_G(v)$. Then necessarily $w \in S$. In the smaller graph $G{-}w$ the set $S \setminus \{w\}$ is a minimum (u,v)-separating set, and the induction hypothesis yields that there are $k-1$ independent paths $u \xrightarrow{\star} v$ in $G{-}w$. Together with the path $u \to w \to v$, there are k independent paths $u \xrightarrow{\star} v$ in G as required.

(2) Assume then that $N_G(u) \cap N_G(v) = \emptyset$, and denote by $H_u = N^*_{G-S}(u)$ and $H_v = N^*_{G-S}(v)$ the connected components of $G{-}S$ for u and v.

(2.1) Suppose next that $S \nsubseteq N_G(u)$ and $S \nsubseteq N_G(v)$.

Let $\widehat{v}$ be a new vertex, and define G_u to be the graph on $H_u \cup S \cup \{\widehat{v}\}$ having the edges of $G[H_u \cup S]$ together with $\widehat{v}w_i$ for all $i \in [1,k]$. The graph G_u is connected and it is smaller than G. Indeed, in order for S to be a minimum separating set, all $w_i \in S$ have to be adjacent to some vertex in H_v. This shows that $\varepsilon_{G_u} \le \varepsilon_G$, and, moreover, the assumption (2.1) rules out the case $H_v = \{v\}$. So $|H_v| \ge 2$ and $\nu_{G_u} < \nu_G$.



If S' is any $(u,\widehat{v})$-separating set of G_u, then S' will separate u from all $w_i \in S \setminus S'$ in G. This means that S' separates u and v in G. Since k is the size of a minimum (u,v)-separating set, we have $|S'| \ge k$. We noted that G_u is smaller than G, and thus by the induction hypothesis, there are k independent paths $u \xrightarrow{\star} \widehat{v}$ in G_u. This is possible only if there exist k paths $u \xrightarrow{\star} w_i$, one for each $i \in [1,k]$, that have only the end u in common.

By the present assumption, also u is nonadjacent to some vertex of S. A symmetric argument applies to the graph G_v (with a new vertex $\widehat{u}$), which is defined similarly to G_u. This yields that there are k paths $w_i \xrightarrow{\star} v$ that have only the end v in common. When we combine these with the above paths $u \xrightarrow{\star} w_i$, we obtain k independent paths $u \xrightarrow{\star} w_i \xrightarrow{\star} v$ in G.

(2.2) There remains the case, where for *all* (u,v)-separating sets S of k elements, either $S \subseteq N_G(u)$ or $S \subseteq N_G(v)$. (Note that then, by (2), $S \cap N_G(v) = \emptyset$ or $S \cap N_G(u) = \emptyset$.)

Let $P = efQ$ be a shortest path $u \xrightarrow{\star} v$ in G, where $e = ux$, $f = xy$, and $Q\colon y \xrightarrow{\star} v$. Notice that, by the assumption (2), $|P| \ge 3$, and so $y \ne v$. In the smaller graph $G{-}f$, let S' be a minimum set that separates u and v.

If $|S'| \ge k$, then, by the induction hypothesis, there are k independent paths $u \xrightarrow{\star} v$ in $G{-}f$. But these are paths of G, and the claim is clear in this case.

If, on the other hand, $|S'| < k$, then u and v are still connected in $G{-}S'$. Every path $u \xrightarrow{\star} v$ in $G{-}S'$ necessarily travels along the edge $f = xy$, and so $x, y \notin S'$.
Let

$$S_x = S' \cup \{x\} \qquad \text{and} \qquad S_y = S' \cup \{y\}\,.$$

These sets separate u and v in G (by the above fact), and they have size k. By our current assumption, the vertices of S_y are adjacent to v, since the path P is shortest and so $uy \notin G$ (meaning that u is not adjacent to all of S_y). The assumption (2) yields that u is adjacent to all of S_x, since $ux \in G$. But now both u and v are adjacent to the vertices of S', which contradicts the assumption (2). □

Theorem 2.8 (Menger **(1927)).** *A graph G is k-connected if and only if every two vertices are connected by at least k independent paths.*

Proof. If any two vertices are connected by k independent paths, then it is clear that $\kappa(G) \geq k$.

In converse, suppose that $\kappa(G) = k$, but that G has vertices u and v connected by at most $k-1$ independent paths. By Theorem 2.7, it must be that $e = uv \in G$. Consider the graph $G{-}e$. Now u and v are connected by at most $k-2$ independent paths in $G{-}e$, and by Theorem 2.7, u and v can be separated in $G{-}e$ by a set S with $|S| = k-2$. Since $\nu_G > k$ (because $\kappa(G) = k$), there exists a $w \in G$ that is not in $S \cup \{u, v\}$. The vertex w is separated in $G{-}e$ by S from u or from v; otherwise there would be a path $u \xrightarrow{\star} v$ in $(G{-}e){-}S$. Say, this vertex is u. The set $S \cup \{v\}$ has $k-1$ elements, and it separates u from w in G, which contradicts the assumption that $\kappa(G) = k$. This proves the claim. □

We state without a proof the corresponding separation property for edge connectivity.

Definition. Let G be a graph. A uv-**disconnecting set** is a set $F \subseteq E_G$ such that every path $u \xrightarrow{\star} v$ contains an edge from F.

Theorem 2.9. *Let $u, v \in G$ with $u \neq v$ in a graph G. Then the maximum number of edge-disjoint paths $u \xrightarrow{\star} v$ equals the minimum number k of edges in a uv-disconnecting set.*

Corollary 2.4. *A graph G is k-edge connected if and only if every two vertices are connected by at least k edge disjoint paths.*

Example 2.9. Recall the definition of the cube Q_k from Example 1.5. We show that $\kappa(Q_k) = k$.

First of all, $\kappa(Q_k) \leq \delta(Q_k) = k$. In converse, we show the claim by induction. Extract from Q_k the disjoint subgraphs: G_0 induced by $\{0u \mid u \in \mathbb{B}^{k-1}\}$ and G_1 induced by $\{1u \mid u \in \mathbb{B}^{k-1}\}$. These are (isomorphic to) Q_{k-1}, and Q_k is obtained from the union of G_0 and G_1 by adding the 2^{k-1} edges $(0u, 1u)$ for all $u \in \mathbb{B}^{k-1}$.

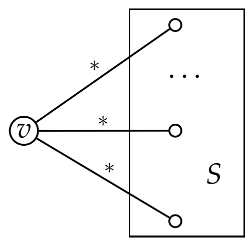
Let S be a separating set of Q_k with $|S| \leq k$. If both G_0-S and G_1-S were connected, also Q_k-S would be connected, since one pair $(0u, 1u)$ necessarily remains in Q_k-S. So we can assume that G_0-S is disconnected. (The case for G_1-S is symmetric.) By the induction hypothesis, $\kappa(G_0) = k-1$, and hence S contains at least $k-1$ vertices of G_0 (and so $|S| \geq k-1$). If there were no vertices from G_1 in S, then, of course, G_1-S is connected, and the edges $(0u, 1u)$ of Q_k would guarantee that Q_k-S is connected; a contradiction. Hence $|S| \geq k$.

Example 2.10. We have $\kappa'(Q_k) = k$ for the k-cube. Indeed, by Whitney's theorem, $\kappa(G) \leq \kappa'(G) \leq \delta(G)$. Since $\kappa(Q_k) = k = \delta(Q_k)$, also $\kappa'(Q_k) = k$.

Algorithmic Problem. The connectivity problems tend to be algorithmically difficult. In the **disjoint paths problem** we are given a set (u_i, v_i) of pairs of vertices for $i = 1, 2, \dots, k$, and it is asked whether there exist paths $P_i \colon u_i \xrightarrow{\star} v_i$ that have no vertices in common. This problem was shown to be NP-complete by KNUTH in 1975. (However, for *fixed* k, the problem has a fast algorithm due to ROBERTSON and SEYMOUR (1986).)

Dirac's fans

DEFINITION. Let $v \in G$ and $S \subseteq V_G$ such that $v \notin S$ in a graph G. A set of paths from v to a vertex in S is called a (v, S)-**fan**, if they have only v in common.



Theorem 2.10 (DIRAC (1960)). *A graph G is k-connected if and only if $\nu_G > k$ and for every $v \in G$ and $S \subseteq V_G$ with $|S| \geq k$ and $v \notin S$, there exists a (v, S)-fan of k paths.*

Proof. Exercise. □

Theorem 2.11 (DIRAC (1960)). *Let G be a k-connected graph for $k \geq 2$. Then for any k vertices, there exists a cycle of G containing them.*

Proof. First of all, since $\kappa(G) \geq 2$, G has no cut vertices, and thus no bridges. It follows that every edge, and thus every vertex of G belongs to a cycle.

Let $S \subseteq V_G$ be such that $|S| = k$, and let C be a cycle of G that contains the maximum number of vertices of S. Let the vertices of $S \cap V_C$ be $v_1, \dots, v_r$ listed in order around C so that each pair (v_i, v_{i+1}) (with indices modulo r) defines a path along C (except in the special case where $r = 1$). Such a path is referred to as a *segment* of C. If C contains all vertices of S, then we are done; otherwise, suppose $v \in S$ is not on C.

It follows from Theorem 2.10 that there is a (v, V_C)-fan of at least $\min\{k, |V_C|\}$ paths. Therefore there are two paths $P \colon v \xrightarrow{\star} u$ and $Q \colon v \xrightarrow{\star} w$ in such a fan that end in the same segment (v_i, v_{i+1}) of C. Then the path $W \colon u \xrightarrow{\star} w$ (or $w \xrightarrow{\star} u$) along C contains all vertices of $S \cap V_C$. But now PWQ^{-1} is a cycle of G that contains v and all v_i for $i \in [1, r]$. This contradicts the choice of C, and proves the claim. □

3

Tours and Matchings

3.1 Eulerian graphs

The first proper problem in graph theory was the Königsberg bridge problem. In general, this problem concerns of travelling in a graph such that one tries to avoid using any edge twice. In practice these eulerian problems occur, for instance, in optimizing distribution networks – such as delivering mail, where in order to save time each street should be travelled only once. The same problem occurs in mechanical graph plotting, where one avoids lifting the pen off the paper while drawing the lines.

Euler tours

DEFINITION. A walk $W = e_1e_2 \dots e_n$ is a **trail**, if $e_i \neq e_j$ for all $i \neq j$. An **Euler trail** of a graph G is a trail that visits every edge once. A connected graph G is **eulerian**, if it has a closed trail containing every edge of G. Such a trail is called an **Euler tour**.

Notice that if $W = e_1e_2 \dots e_n$ is an Euler tour (and so $E_G = \{e_1, e_2, \dots, e_n\}$), also $e_ie_{i+1} \dots e_ne_1 \dots e_{i-1}$ is an Euler tour for all $i \in [1, n]$. A complete proof of the following **Euler's Theorem** was first given by HIERHOLZER in 1873.

Theorem 3.1 (EULER **(1736),** HIERHOLZER **(1873)).** *A connected graph G is eulerian if and only if every vertex has an even degree.*

Proof. ($\Rightarrow$) Suppose $W\colon u \xrightarrow{\star} u$ is an Euler tour. Let v ($\neq u$) be a vertex that occurs k times in W. Every time an edge arrives at v, another edge departs from v, and therefore $d_G(v) = 2k$. Also, $d_G(u)$ is even, since W starts and ends at u.

($\Leftarrow$) Assume G is a nontrivial connected graph such that $d_G(v)$ is even for all $v \in G$. Let

$$W = e_1e_2 \dots e_n \colon v_0 \xrightarrow{\star} v_n \quad \text{with} \quad e_i = v_{i-1}v_i$$

be a longest trail in G. It follows that all $e = v_nw \in G$ are among the edges of W, for, otherwise, W could be prolonged to We. In particular, $v_0 = v_n$, that is, W is a closed trail. (Indeed, if it were $v_n \neq v_0$ and v_n occurs k times in W, then $d_G(v_n) = 2(k-1)+1$ and that would be odd.)

If W is not an Euler tour, then, since G is connected, there exists an edge $f = v_iu \in G$ for some i, which is not in W. However, now

$$e_{i+1} \dots e_ne_1 \dots e_if$$

is a trail in G, and it is longer than W. This contradiction to the choice of W proves the claim. □

Example 3.1. The k-cube Q_k is eulerian for even integers k, because Q_k is k-regular.

Theorem 3.2. *A connected graph has an Euler trail if and only if it has at most two vertices of odd degree.*

Proof. If G has an Euler trail $u \xrightarrow{\star} v$, then, as in the proof of Theorem 3.1, each vertex $w \notin \{u, v\}$ has an even degree.

Assume then that G is connected and has at most two vertices of odd degree. If G has no vertices of odd degree then, by Theorem 3.1, G has an Euler trail. Otherwise, by the handshaking lemma, every graph has an even number of vertices with odd degree, and therefore G has exactly two such vertices, say u and v. Let H be a graph obtained from G by adding a vertex w, and the edges uw and vw. In H every vertex has an even degree, and hence it has an Euler tour, say $u \xrightarrow{\star} v \to w \to u$. Here the beginning part $u \xrightarrow{\star} v$ is an Euler trail of G. □

The Chinese postman

The following problem is due to GUAN MEIGU (1962). Consider a village, where a postman wishes to plan his route to save the legs, but still every street has to be walked through. This problem is akin to Euler's problem and to the shortest path problem.

Let G be a graph with a weight function $\alpha \colon E_G \to \mathbb{R}^+$. The **Chinese postman problem** is to find a minimum weighted tour in G (starting from a given vertex, the post office).

If G is *eulerian*, then any Euler tour will do as a solution, because such a tour traverses each edge exactly once and this is the best one can do. In this case the weight of the optimal tour is the total weight of the graph G, and there is a good algorithm for finding such a tour:

Fleury's algorithm:

- Let $v_0 \in G$ be a chosen vertex, and let W_0 be the trivial path on v_0.
- Repeat the following procedure for $i = 1, 2, \dots$ as long as possible: suppose a trail $W_i = e_1 e_2 \dots e_i$ has been constructed, where $e_j = v_{j-1}v_j$.
  Choose an edge e_{i+1} ($\neq e_j$ for $j \in [1, i]$) so that

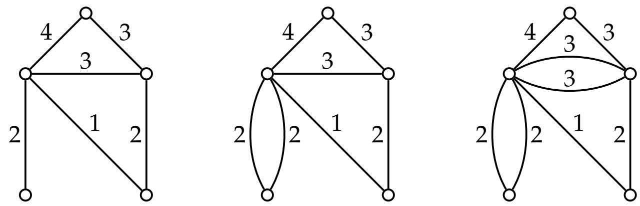
(i) e_{i+1} has an end v_i, and
(ii) e_{i+1} is not a bridge of $G_i = G - \{e_1, \dots, e_i\}$, unless there is no alternative.

Notice that, as is natural, the weights $\alpha(e)$ play no role in the eulerian case.

Theorem 3.3. *If G is eulerian, then any trail of G constructed by Fleury's algorithm is an Euler tour of G.*

Proof. Exercise. □

If G is *not eulerian*, the poor postman has to walk at least one street twice. This happens, *e.g.*, if one of the streets is a dead end, and in general if there is a street corner of an odd number of streets. We can attack this case by reducing it to the eulerian case as follows. An edge $e = uv$ will be **duplicated**, if it is added to G parallel to an existing edge $e' = uv$ with the same weight, $\alpha(e') = \alpha(e)$.



Above we have duplicated two edges. The rightmost multigraph is eulerian.

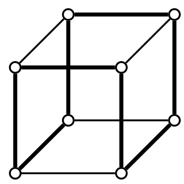
There is a good algorithm by EDMONDS AND JOHNSON (1973) for the construction of an optimal eulerian supergraph by duplications. Unfortunately, this algorithm is somewhat complicated, and we shall skip it.

3.2 Hamiltonian graphs

In the connector problem we reduced the cost of a spanning graph to its minimum. There are different problems, where the cost is measured by an active user of the graph. For instance, in the **travelling salesman problem** a person is supposed to visit each town in his district, and this he should do in such a way that saves time and money. Obviously, he should plan the travel so as to visit each town once, and so that the overall flight time is as short as possible. In terms of graphs, he is looking for a minimum weighted Hamilton cycle of a graph, the vertices of which are the towns and the weights on the edges are the flight times. Unlike for the shortest path and the connector problems no efficient reliable algorithm is known for the travelling salesman problem. Indeed, it is widely believed that no practical algorithm exists for this problem.

Hamilton cycles

DEFINITION. A path P of a graph G is a **Hamilton path**, if P visits every vertex of G once. Similarly, a cycle C is a **Hamilton cycle**, if it visits each vertex once. A graph is **hamiltonian**, if it has a Hamilton cycle.



Note that if $C : u_1 \to u_2 \to \cdots \to u_n$ is a Hamilton cycle, then so is $u_i \to \dots u_n \to u_1 \to \dots u_{i-1}$ for each $i \in [1, n]$, and thus we can choose where to start the cycle.

Example 3.2. It is obvious that each K_n is hamiltonian whenever $n \geq 3$. Also, as is easily seen, $K_{n,m}$ is hamiltonian if and only if $n = m \geq 2$. Indeed, let $K_{n,m}$ have a bipartition (X, Y), where $|X| = n$ and $|Y| = m$. Now, each cycle in $K_{n,m}$ has even length as the graph is bipartite, and thus the cycle visits the sets X, Y equally many times, since X and Y are stable subsets. But then necessarily $|X| = |Y|$.

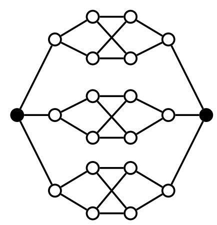
Unlike for eulerian graphs (Theorem 3.1) no good characterization is known for hamiltonian graphs. Indeed, the problem to determine if G is hamiltonian is NP-complete. There are, however, some interesting general conditions.

Lemma 3.1. *If G is hamiltonian, then for every nonempty subset* $S \subseteq V_G$,

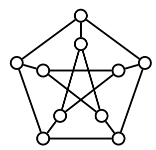
$$c(G-S) \le |S| .$$

Proof. Let $\emptyset \neq S \subseteq V_G$, $u \in S$, and let $C\colon u \xrightarrow{\star} u$ be a Hamilton cycle of G. Assume $G-S$ has k connected components, G_i, $i \in [1, k]$. The case $k = 1$ is trivial, and hence suppose that $k > 1$. Let u_i be the last vertex of C that belongs to G_i, and let v_i be the vertex that follows u_i in C. Now $v_i \in S$ for each i by the choice of u_i, and $v_j \neq v_t$ for all $j \neq t$, because C is a cycle and $u_i v_i \in G$ for all i. Thus $|S| \ge k$ as required. □

Example 3.3. Consider the graph on the right. In G, $c(G-S) = 3 > 2 = |S|$ for the set S of black vertices. Therefore G does not satisfy the condition of Lemma 3.1, and hence it is not hamiltonian. Interestingly this graph is (X, Y)-bipartite of even order with $|X| = |Y|$. It is also 3-regular.



Example 3.4. Consider the **Petersen graph** on the right, which appears in many places in graph theory as a counter example for various conditions. This graph is not hamiltonian, but it does satisfy the condition $c(G-S) \le |S|$ for all $S \neq \emptyset$. Therefore the conclusion of Lemma 3.1 is *not sufficient* to ensure that a graph is hamiltonian.



The following theorem, due to ORE, generalizes an earlier result by DIRAC (1952).

Theorem 3.4 (ORE (1962)). *Let G be a graph of order* $\nu_G \ge 3$, *and let* $u, v \in G$ *be such that*

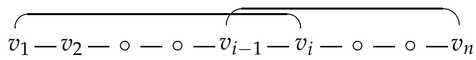
$$d_G(u) + d_G(v) \ge \nu_G .$$

Then G is hamiltonian if and only if $G + uv$ *is hamiltonian.*

Proof. Denote $n = \nu_G$. Let $u, v \in G$ be such that $d_G(u) + d_G(v) \ge n$. If $uv \in G$, then there is nothing to prove. Assume thus that $uv \notin G$.

($\Rightarrow$) This is trivial since if G has a Hamilton cycle C, then C is also a Hamilton cycle of $G + uv$.

($\Leftarrow$) Denote $e = uv$ and suppose that $G + e$ has a Hamilton cycle C. If C does not use the edge e, then it is a Hamilton cycle of G. Suppose thus that e is on C. We may then assume that $C\colon u \xrightarrow{\star} v \to u$. Now $u = v_1 \to v_2 \to \ldots \to v_n = v$ is a Hamilton *path* of G. There exists an i with $1 < i < n$ such that $uv_i \in G$ and $v_{i-1}v \in G$. For, otherwise, $d_G(v) < n - d_G(u)$ would contradict the assumption.



But now $u = v_1 \xrightarrow{\star} v_{i-1} \to v_n \to v_{n-1} \xrightarrow{\star} v_{i+1} \to v_i \to v_1 = u$ is a Hamilton cycle in G. □

Closure

DEFINITION. For a graph G, define inductively a sequence $G_0, G_1, \dots, G_k$ of graphs such that

$$G_0 = G \text{ and } G_{i+1} = G_i + uv\,,$$

where u and v are any vertices such that $uv \notin G_i$ and $d_{G_i}(u) + d_{G_i}(v) \geq \nu_G$. This procedure stops when no new edges can be added to G_k for some k, that is, in G_k, for all $u, v \in G$ either $uv \in G_k$ or $d_{G_k}(u) + d_{G_k}(v) < \nu_G$. The result of this procedure is the **closure** of G, and it is denoted by $cl(G)$ $(= G_k)$.

In each step of the construction of $cl(G)$ there are usually alternatives which edge uv is to be added to the graph, and therefore the above procedure is not deterministic. However, the *final result* $cl(G)$ is independent of the choices.

Lemma 3.2. *The closure $cl(G)$ is uniquely defined for all graphs G of order $\nu_G \geq 3$.*

Proof. Denote $n = \nu_G$. Suppose there are two ways to close G, say

$$H = G + \{e_1, \dots, e_r\} \text{ and } H' = G + \{f_1, \dots, f_s\}\,,$$

where the edges are added in the given orders. Let $H_i = G + \{e_1, \dots, e_i\}$ and $H'_i = G + \{f_1, \dots, f_i\}$. For the initial values, we have $G = H_0 = H'_0$. Let $e_k = uv$ be the first edge such that $e_k \neq f_i$ for all i. Then $d_{H_{k-1}}(u) + d_{H_{k-1}}(v) \geq n$, since $e_k \in H_k$, but $e_k \notin H_{k-1}$. By the choice of e_k, we have $H_{k-1} \subseteq H'$, and thus also $d_{H'}(u) + d_{H'}(v) \geq n$, which means that $e = uv$ must be in H'; a contradiction. Therefore $H \subseteq H'$. Symmetrically, we deduce that $H' \subseteq H$, and hence $H' = H$. □

Theorem 3.5. *Let G be a graph of order $\nu_G \geq 3$.*

(i) G is hamiltonian if and only if its closure $cl(G)$ is hamiltonian.
(ii) If $cl(G)$ is a complete graph, then G is hamiltonian.

Proof. First, $G \subseteq cl(G)$ and G spans $cl(G)$, and thus if G is hamiltonian, so is $cl(G)$.

In the other direction, let $G = G_0, G_1, \dots, G_k = cl(G)$ be a construction sequence of the closure of G. If $cl(G)$ is hamiltonian, then so are $G_{k-1}, \dots, G_1$ and G_0 by Theorem 3.4.

The Claim (ii) follows from (i), since each complete graph is hamiltonian. □

Theorem 3.6. *Let G be a graph of order $\nu_G \geq 3$. Suppose that for all nonadjacent vertices u and v, $d_G(u) + d_G(v) \geq \nu_G$. Then G is hamiltonian. In particular, if $\delta(G) \geq \frac{1}{2}\nu_G$, then G is hamiltonian.*

Proof. Since $d_G(u) + d_G(v) \geq \nu_G$ for all nonadjacent vertices, we have $cl(G) = K_n$ for $n = \nu_G$, and thus G is hamiltonian. The second claim is immediate, since now $d_G(u) + d_G(v) \geq \nu_G$ for all $u, v \in G$ whether adjacent or not. □

Chvátal's condition

The hamiltonian problem of graphs has attracted much attention, at least partly because the problem has practical significance. (Indeed, the first example where DNA computing was applied, was the hamiltonian problem.)

There are some general improvements of the previous results of this chapter, and quite many improvements in various special cases, where the graphs are somehow restricted. We become satisfied by two general results.

Theorem 3.7 (CHVÁTAL **(1972)).** *Let G be a graph with $V_G = \{v_1, v_2, \ldots, v_n\}$, for $n \geq 3$, ordered so that $d_1 \leq d_2 \leq \cdots \leq d_n$, for $d_i = d_G(v_i)$. If for every $i < n/2$,*

$$d_i \leq i \implies d_{n-i} \geq n - i\,, \tag{3.1}$$

then G is hamiltonian.

Proof. First of all, we may suppose that G is closed, $G = cl(G)$, because G is hamiltonian if and only if $cl(G)$ is hamiltonian, and adding edges to G does not decrease any of its degrees, that is, if G satisfies (3.1), so does $G + e$ for every e. We show that, in this case, $G = K_n$, and thus G is hamiltonian.

Assume on the contrary that $G \neq K_n$, and let $uv \notin G$ with $d_G(u) \leq d_G(v)$ be such that $d_G(u) + d_G(v)$ is as large as possible. Because G is closed, we must have $d_G(u) + d_G(v) < n$, and therefore $d_G(u) = i < n/2$. Let $A = \{w \mid vw \notin G, w \neq v\}$. By our choice, $d_G(w) \leq i$ for all $w \in A$, and, moreover,

$$|A| = (n-1) - d_G(v) \geq d_G(u) = i\,.$$

Consequently, there are at least i vertices w with $d_G(w) \leq i$, and so $d_i \leq d_G(u) = i$.

Similarly, for each vertex from $B = \{w \mid uw \notin G, w \neq u\}$, $d_G(w) \leq d_G(v) < n - d_G(u) = n - i$, and

$$|B| = (n-1) - d_G(u) = (n-1) - i\,.$$

Also $d_G(u) < n - i$, and thus there are at least $n - i$ vertices w with $d_G(w) < n - i$. Consequently, $d_{n-i} < n - i$. This contradicts the obtained bound $d_i \leq i$ and the condition (3.1). □

Note that the condition (3.1) is easily checkable for any given graph.

3.3 Matchings

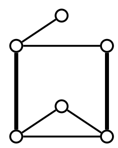
In matching problems we are given an availability relation between the elements of a set. The problem is then to find a pairing of the elements so that each element is paired (matched) uniquely with an available companion.

A special case of the matching problem is the **marriage problem**, which is stated as follows. Given a set X of boys and a set Y of girls, under what condition can each boy marry a girl who cares to marry him? This problem has many variations. One of them is the **job assignment problem**, where we are given n applicants and m jobs, and we should assign each applicant to a job he is qualified. The problem is that an applicant may be qualified for several jobs, and a job may be suited for several applicants.

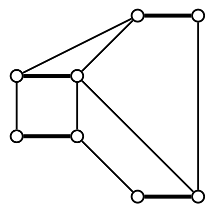
Maximum matchings

DEFINITION. For a graph G, a subset $M \subseteq E_G$ is a **matching** of G, if M contains no adjacent edges. The two ends of an edge $e \in M$ are **matched under** M. A matching M is a **maximum matching**, if for no matching M', $|M| < |M'|$.

The two vertical edges on the right constitute a matching M that is *not a maximum matching*, although you cannot add any edges to M to form a larger matching. This matching is not maximum because the graph has a matching of three edges.

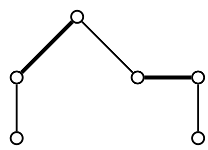


DEFINITION. A matching M **saturates** $v \in G$, if v is an end of an edge in M. Also, M **saturates** $A \subseteq V_G$, if it saturates every $v \in A$. If M saturates V_G, then M is a **perfect matching**.



It is clear that every perfect matching is maximum. On the right the horizontal edges form a perfect matching.

DEFINITION. Let M be a matching of G. An odd path $P = e_1 e_2 \dots e_{2k+1}$ is M-**augmented**, if



- P alternates between $E_G \setminus M$ and M (that is, $e_{2i+1} \in G - M$ and $e_{2i} \in M$), and
- the ends of P are not saturated.

Lemma 3.3. *If G is connected with $\Delta(G) \leq 2$, then G is a path or a cycle.*

Proof. Exercise. □

We start with a result that gives a necessary and sufficient condition for a matching to be maximum. One can use the first part of the proof to construct a maximum matching in an iterative manner starting from any matching M and from any M-augmented path.

Theorem 3.8 (BERGE **(1957)).** *A matching M of G is a maximum matching if and only if there are no M-augmented paths in G.*

Proof. ($\Rightarrow$) Let a matching M have an M-augmented path $P = e_1e_2 \dots e_{2k+1}$ in G. Here $e_2, e_4, \dots, e_{2k} \in M$, $e_1, e_3, \dots, e_{2k+1} \notin M$. Define $N \subseteq E_G$ by

$$N = (M \setminus \{e_{2i} \mid i \in [1,k]\}) \cup \{e_{2i+1} \mid i \in [0,k]\}.$$

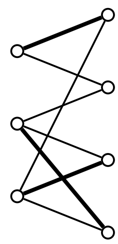
Now, N is a matching of G, and $|N| = |M| + 1$. Therefore M is not a maximum matching.

($\Leftarrow$) Assume N is a maximum matching, but M is not. Hence $|N| > |M|$. Consider the subgraph $H = G[M \triangle N]$ for the symmetric difference $M \triangle N$. We have $d_H(v) \leq 2$ for each $v \in H$, because v is an end of at most one edge in M and N. By Lemma 3.3, each connected component A of H is either a path or a cycle.

Since no $v \in A$ can be an end of two edges from N or from M, each connected component (path or a cycle) A alternates between N and M. Now, since $|N| > |M|$, there is a connected component A of H, which has more edges from N than from M. This A cannot be a cycle, because an alternating cycle has even length, and it thus contains equally many edges from N and M. Hence $A\colon u \xrightarrow{\star} v$ is a path (of odd length), which starts and ends with an edge from N. Because A is a connected component of H, the ends u and v are not saturated by M, and, consequently, A is an M-augmented path. This proves the theorem. □

Example 3.5. Consider the k-cube Q_k for $k \geq 1$. Each maximum matching of Q_k has 2^{k-1} edges. Indeed, the matching $M = \{(0u, 1u) \mid u \in \mathbb{B}^{k-1}\}$, has 2^{k-1} edges, and it is clearly perfect.

Hall's theorem



For a subset $S \subseteq V_G$ of a graph G, denote

$$N_G(S) = \{v \mid uv \in G \text{ for some } u \in S\}.$$

If G is (X,Y)-bipartite, and $S \subseteq X$, then $N_G(S) \subseteq Y$.

The following result, known as the

Theorem 3.9 (HALL **(1935)).** *Let G be a (X,Y)-bipartite graph. Then G contains a matching M saturating X if and only if*

$$|S| \leq |N_G(S)| \quad \textit{for all } S \subseteq X. \tag{3.2}$$

Proof. ($\Rightarrow$) Let M be a matching that saturates X. If $|S| > |N_G(S)|$ for some $S \subseteq X$, then not all $x \in S$ can be matched with different $y \in N_G(S)$.

($\Leftarrow$) Let G satisfy Hall's condition (3.2). We prove the claim by induction on $|X|$.

If $|X| = 1$, then the claim is clear. Let then $|X| \geq 2$, and assume (3.2) implies the existence of a matching that saturates every proper subset of X.

If $|N_G(S)| \geq |S| + 1$ for every nonempty $S \subseteq X$ with $S \neq X$, then choose an edge $uv \in G$ with $u \in X$, and consider the induced subgraph $H = G - \{u, v\}$. For all $S \subseteq X \setminus \{u\}$, $|N_H(S)| \geq |N_G(S)| - 1 \geq |S|$, and hence, by the induction hypothesis, H contains a matching M saturating $X \setminus \{u\}$. Now $M \cup \{uv\}$ is a matching saturating X in G, as was required.

Suppose then that there exists a nonempty subset $R \subseteq X$ with $R \neq X$ such that $|N_G(R)| = |R|$. The induced subgraph $H_1 = G[R \cup N_G(R)]$ satisfies (3.2) (since G does), and hence, by the induction hypothesis, H_1 contains a matching M_1 that saturates R (with the other ends in $N_G(R)$).

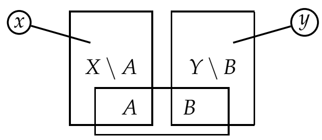
Also, the induced subgraph $H_2 = G[V_G \setminus A]$, for $A = R \cup N_G(R)$, satisfies (3.2). Indeed, if there were a subset $S \subseteq X \setminus R$ such that $|N_{H_2}(S)| < |S|$, then we would have

$$|N_G(S \cup R)| = |N_{H_2}(S)| + |N_{H_1}(R)| < |S| + |N_G(R)| = |S| + |R| = |S \cup R|$$

(since $S \cap R = \emptyset$), which contradicts (3.2) for G. By the induction hypothesis, H_2 has a matching M_2 that saturates $X \setminus R$ (with the other ends in $Y \setminus N_G(R)$). Combining the matchings for H_1 and H_2, we get a matching $M_1 \cup M_2$ saturating X in G. □

Second proof. This proof of the direction ($\Leftarrow$) uses Menger's theorem. Let H be the graph obtained from G by adding two new vertices x, y such that x is adjacent to each $v \in X$ and y is adjacent to each $v \in Y$. There exists a matching saturating X if (and only if) the number of independent paths $x \xrightarrow{\star} y$ is equal to $|X|$. For this, by Menger's theorem, it suffices to show that every set S that separates x and y in H has at least $|X|$ vertices.

Let $S = A \cup B$, where $A \subseteq X$ and $B \subseteq Y$. Now, vertices in $X \setminus A$ are not adjacent to vertices of $Y \setminus B$, and hence we have $N_G(X \setminus A) \subseteq B$, and thus that $|X \setminus A| \leq |N_G(X \setminus A)| \leq |B|$ using the condition (3.2).



We conclude that $|S| = |A| + |B| \geq |X|$. □

Corollary 3.1 (FROBENIUS (1917)). *If G is a k-regular bipartite graph with $k > 0$, then G has a perfect matching.*

Proof. Let G be k-regular (X, Y)-bipartite graph. By regularity, $k \cdot |X| = \varepsilon_G = k \cdot |Y|$, and hence $|X| = |Y|$. Let $S \subseteq X$. Denote by E_1 the set of the edges with an end in S, and by E_2 the set of the edges with an end in $N_G(S)$. Clearly, $E_1 \subseteq E_2$. Therefore, $k \cdot |N_G(S)| = |E_2| \geq |E_1| = k \cdot |S|$, and so $|N_G(S)| \geq |S|$. By Theorem 3.9, G has a matching that saturates X. Since $|X| = |Y|$, this matching is necessarily perfect. □

Applications of Hall's theorem

DEFINITION. Let $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ be a family of finite nonempty subsets of a set S. (S_i need not be distinct.) A **transversal** (or a **system of distinct representatives**) of $\mathcal{S}$ is a subset $T \subseteq S$ of m distinct elements one from each S_i.

As an example, let $S = [1,6]$, and let $S_1 = S_2 = \{1,2\}$, $S_3 = \{2,3\}$ and $S_4 = \{1,4,5,6\}$. For $\mathcal{S} = \{S_1, S_2, S_3, S_4\}$, the set $T = \{1,2,3,4\}$ is a transversal. If we add the set $S_5 = \{2,3\}$ to $\mathcal{S}$, then it is impossible to find a transversal for this new family.

The connection of transversals to the Marriage Theorem is as follows. Let $S = Y$ and $X = [1,m]$. Form an (X,Y)-bipartite graph G such that there is an edge (i,s) if and only if $s \in S_i$. The possible transversals T of $\mathcal{S}$ are then obtained from the matchings M saturating X in G by taking the ends in Y of the edges of M.

Corollary 3.2. *Let $\mathcal{S}$ be a family of finite nonempty sets. Then $\mathcal{S}$ has a transversal if and only if the union of any k of the subsets S_i of $\mathcal{S}$ contains at least k elements.*

Example 3.6. An $m \times n$ **latin rectangle** is an $m \times n$ integer matrix M with entries $M_{ij} \in [1,n]$ such that the entries in the same row and in the same column are different. Moreover, if $m = n$, then M is a **latin square**. Note that in a $m \times n$ latin rectangle M, we always have that $m \leq n$.

We show the following: *Let M be an $m \times n$ latin rectangle (with $m < n$). Then M can be extended to a latin square by the addition of $n - m$ new rows.*

The claim follows when we show that M can be extended to an $(m+1) \times n$ latin rectangle. Let $A_i \subseteq [1,n]$ be the set of those elements that do not occur in the i-th column of M. Clearly, $|A_i| = n - m$ for each i, and hence $\sum_{i \in I} |A_i| = |I|(n-m)$ for all subsets $I \subseteq [1,n]$. Now $|\cup_{i \in I} A_i| \geq |I|$, since otherwise at least one element from the union would be in more than $n - m$ of the sets A_i with $i \in I$. However, each row has all the n elements, and therefore each i is missing from exactly $n - m$ columns. By Marriage Theorem, the family $\{A_1, A_2, \dots, A_n\}$ has a transversal, and this transversal can be added as a new row to M. This proves the claim.

Tutte's theorem

The next theorem is a classic characterization of perfect matchings.

DEFINITION. A connected component of a graph G is said to be **odd** (**even**), if it has an odd (even) number of vertices. Denote by $c_{\text{odd}}(G)$ the number of odd connected components in G.

Denote by $m(G)$ be the number of edges in a maximum matching of a graph G.

Theorem 3.10 (Tutte-Berge Formula). *Each maximum matching of a graph G has*

$$m(G) = \min_{S \subseteq V_G} \frac{\nu_G + |S| - c_{\text{odd}}(G{-}S)}{2} \tag{3.3}$$

elements.

Note that the condition in (ii) includes the case, where $S = \emptyset$.

Proof. We prove the result for connected graphs. The result then follows for disconnected graphs by adding the formulas for the connected components.

We observe first that $\leq$ holds in (3.3), since, for all $S \subseteq V_G$,

$$m(G) \leq |S| + m(G{-}S) \leq |S| + \frac{|V_G \setminus S| - c_{\text{odd}}(G{-}S)}{2} = \frac{\nu_G + |S| - c_{\text{odd}}(G{-}S)}{2}.$$

Indeed, each odd component of $G{-}S$ must have at least one unsaturated vertex.

The proof proceeds by induction on ν_G. If $\nu_G = 1$, then the claim is trivial. Suppose that $\nu_G \geq 2$.

Assume first that there exists a vertex $v \in G$ such that v is saturated by all maximum matchings. Then $m(G{-}v) = m(G) - 1$. For a subset $S' \subseteq G{-}v$, denote $S = S' \cup \{v\}$. By the induction hypothesis, for all $S' \subseteq G{-}v$,

$$\begin{aligned} m(G) - 1 &\geq \frac{1}{2}\left((\nu_G - 1) + |S'| - c_{\text{odd}}(G{-}(S' \cup \{v\}))\right) \\ &= \frac{1}{2}\left((\nu_G + |S| - c_{\text{odd}}(G{-}S))\right) - 1. \end{aligned}$$

The claim follows from this.

Suppose then that for each vertex v, there is a maximum matching that does not saturate v. We claim that $m(G) = (\nu_G - 1)/2$. Suppose to the contrary, and let M be a maximum matching having two different unsaturated vertices u and v, and choose M so that the distance $d_G(u,v)$ is as small as possible. Now $d_G(u,v) \geq 2$, since otherwise $uv \in G$ could be added to M, contradicting the maximality of M. Let w be an intermediate vertex on a shortest path $u \xrightarrow{\star} v$. By assumption, there exists a maximum matching N that does not saturate w. We can choose N such that the intersection $M \cap N$ is maximal. Since $d_G(u,w) < d_G(u,v)$ and $d_G(w,v) < d_G(u,v)$, N saturates both u and v. The (maximum) matchings N and M leave equally many vertices unsaturated, and hence there exists another vertex $x \neq w$ saturated by M but which is unsaturated by N. Let $e = xy \in M$. If y is also unsaturated by N, then $N \cup \{e\}$ is a matching, contradicting maximality of N. It also follows that $y \neq w$. Therefore there exists an edge $e' = yz$ in N, where $z \neq x$. But now $N' = N \cup \{e\} \setminus \{e'\}$ is a maximum matching that does not saturate w. However, $N \cap M \subset N' \cap M$ contradicts the choice of N. Therefore, every maximum matching leaves exactly one vertex unsaturated, i.e., $m(G) = (\nu_G - 1)/2$.

In this case, for $S = \emptyset$, the right hand side of (3.3) gets value $(\nu_G - 1)/2$, and hence, by the beginning of the proof, this must be the minimum of the right hand side. □

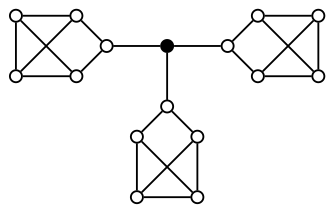
For perfect matchings we have the following corollary, since for a perfect matching we have $m(G) = (1/2)\nu_G$.

Theorem 3.11 (TUTTE (1947)). *Let G be a nontrivial graph. The following are equivalent.*

(i) G has a perfect matching.
(ii) For every proper subset $S \subset V_G$, $c_{\text{odd}}(G-S) \le |S|$.

Tutte's theorem does not provide a good algorithm for constructing a perfect matching, because the theorem requires 'too many cases'. Its applications are mainly in the proofs of other results that are related to matchings. There is a good algorithm due to EDMONDS (1965), which uses 'blossom shrinkings', but this algorithm is somewhat involved.

Example 3.7. The simplest connected graph that has no perfect matching is the path P_3. Here removing the middle vertex creates two odd components.



The next 3-regular graph (known as the **Sylvester graph**) does not have a perfect matching, because removing the black vertex results in a graph with three odd connected components. This graph is the smallest regular graph with an odd degree that has no perfect matching.

Using Theorem 3.11 we can give a short proof of PETERSEN's result for 3-regular graphs (1891).

Theorem 3.12 (PETERSEN (1891)). *If G is a bridgeless 3-regular graph, then it has a perfect matching.*

Proof. Let S be a proper subset of V_G, and let G_i, $i \in [1,t]$, be the odd connected components of $G-S$. Denote by m_i the number of edges with one end in G_i and the other in S. Since G is 3-regular,

$$\sum_{v \in G_i} d_G(v) = 3 \cdot \nu_{G_i} \quad \text{and} \quad \sum_{v \in S} d_G(v) = 3 \cdot |S| \, .$$

The first of these implies that

$$m_i = \sum_{v \in G_i} d_G(v) - 2 \cdot \varepsilon_{G_i}$$

is odd. Furthermore, $m_i \neq 1$, because G has no bridges, and therefore $m_i \geq 3$. Hence the number of odd connected components of $G-S$ satisfies

$$t \le \frac{1}{3}\sum_{i=1}^{t} m_i \le \frac{1}{3}\sum_{v \in S} d_G(v) = |S| \, ,$$

and so, by Theorem 3.11, G has a perfect matching. □

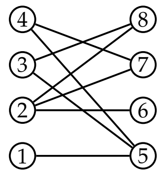
Stable Marriages

DEFINITION. Consider a bipartite graph G with a bipartition (X, Y) of the vertex set. In addition, each vertex $x \in G$ supplies an order of preferences of the vertices of $N_G(x)$. We write $u <_x v$, if x prefers v to u. (Here $u, v \in Y$, if $x \in X$, and $u, v \in X$, if $x \in Y$.) A matching M of G is said to be **stable**, if for each unmatched pair $xy \notin M$ (with $x \in X$ and $y \in Y$), it is not the case that x and y prefer each other better than their matched companions:

$$xv \in M \text{ and } y <_x v, \text{ or } uy \in M \text{ and } x <_y u.$$

We omit the proof of the next theorem.

Theorem 3.13. *For bipartite graphs G, a stable matching exists for all lists of preferences.*



Example 3.8. That was the good news. There is a catch, of course. A stable matching need not saturate X and Y. For instance, the graph on the right does have a perfect matching (of 4 edges).

Suppose the preferences are the following:

$1\colon 5$	$2\colon 6 < 8 < 7$	$3\colon 8 < 5$	$4\colon 7 < 5$
$5\colon 4 < 1 < 3$	$6\colon 2$	$7\colon 2 < 4$	$8\colon 3 < 2$

Then there is no stable matchings of four edges. A stable matching of G is the following: $M = \{28, 35, 47\}$, which leaves 1 and 6 unmatched. (You should check that there is no stable matching containing the edges 15 and 26.)

Theorem 3.14. *Let $G = K_{n,n}$ be a complete bipartite graph. Then G has a perfect and stable matching for all lists of preferences.*

Proof. Let the bipartition be (X, Y). The algorithm by GALE AND SHAPLEY (1962) works as follows.

Procedure.
Set $M_0 = \emptyset$, and $P(x) = \emptyset$ for all $x \in X$.
Then iterate the following process until all vertices are saturated:
Choose a vertex $x \in X$ that is unsaturated in M_{i-1}. Let $y \in Y$ be the most preferred vertex for x such that $y \notin P(x)$.

(1) Add y to $P(x)$.
(2) If y is not saturated, then set $M_i = M_{i-1} \cup \{xy\}$.
(3) If $zy \in M_{i-1}$ and $z <_y x$, then set $M_i = (M_{i-1} \setminus \{zy\}) \cup \{xy\}$.

First of all, the procedure terminates, since a vertex $x \in X$ takes part in the iteration at most n times (once for each $y \in Y$). The final outcome, say $M = M_t$, is a perfect matching, since the iteration continues until there are no unsaturated vertices $x \in X$.

Also, the matching $M = M_t$ is stable. Note first that, by (3), if $xy \in M_i$ and $zy \in M_j$ for some $x \neq z$ and $i < j$, then $x <_y z$. Assume the that $xy \in M$, but $y <_x z$ for some $z \in Y$. Then xy is added to the matching at some step, $xy \in M_i$, which means that $z \in P(x)$ at this step (otherwise x would have 'proposed' z). Hence x took part in the iteration at an earlier step M_k, $k < i$ (where z was put to the list $P(x)$, but xz was not added). Thus, for some $u \in X$, $uz \in M_{k-1}$ and $x <_z u$, and so in M the vertex z is matched to some w with $x <_z w$.

Similarly, if $x <_y v$ for some $v \in X$, then $y <_v z$ for the vertex $z \in Y$ such that $vz \in M$. □