SAPIENZA
UNIVERSITÀ DI ROMA

"Sapienza" University of Rome
Faculty of Information Engineering,
Informatics and Statistics
Department of Computer Science

Graph Theory

Lecture notes integrated with the book "Graph Theory", R. Diestel

Author
Simone Bianco

April 2, 2025

Contents

Information and Contacts

Personal notes and summaries collected as part of the *Graph Theory* course offered by the degree in Computer Science of the University of Rome "La Sapienza".

Further information and notes can be found at the following link:
https://github.com/Exyss/university-notes. Anyone can feel free to report inaccuracies, improvements or requests through the Issue system provided by GitHub itself or by contacting the author privately:

- Email: **bianco.simone@outlook.it**
- LinkedIn: **Simone Bianco**

The notes are constantly being updated, so please check if the changes have already been made in the most recent version.

Suggested prerequisites:

Sufficient knowledge of algorithm design, probability and combinatorics

Licence:

1

Introduction to Graph Theory

1.1 Graphs, subgraphs and neighbors

The city of Königsberg (in Prussia), situated along the Pregel River, is divided into four regions: two parts of the mainland and two islands, Kneiphof and Lomse. In the 18th century, these areas were connected by *seven bridges* that spanned the river, crossing it in various directions. As time passed, a fascinating question emerged among residents: could one walk through the city, crossing each of the seven bridges exactly once, and return to the starting point? This became known as the *Seven Bridges of Königsberg problem*.

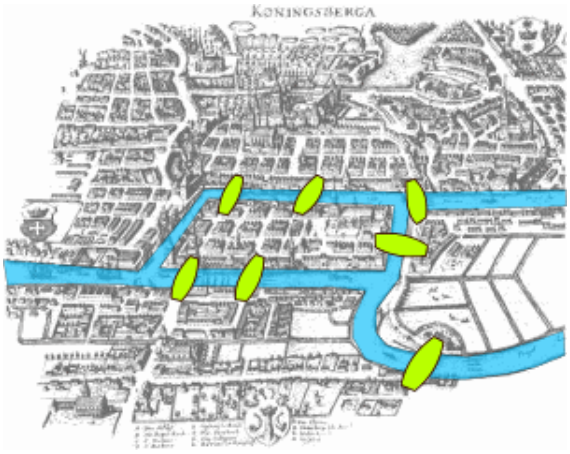


Figure 1.1: The map of Königsberg in the 18th century, showing the layout of the seven bridges.

Though it appeared to be a simple challenge, no one had succeeded in solving it. The riddle became a popular topic of conversation, discussed in markets and taverns alike. Some believed the walk was possible with the right path, while others doubted it could be done. Word of this puzzling problem eventually reached the brilliant Swiss mathematician,

Leonhard Euler. Fascinated by the challenge, Euler sought a solution – not by walking the streets himself, but by abstracting the problem into a more general form. Euler realized that the precise layout of the city itself wasn't essential. What really mattered was how the landmasses were **connected** by the bridges. He began by representing each landmass as a dot and each bridge as a line between them. By doing this, he removed unnecessary details and created a simple yet powerful combinatorial structure, which we now call a **graph**.

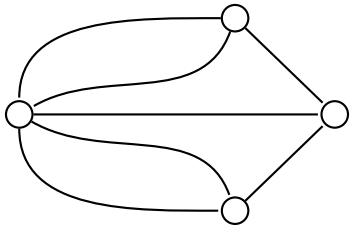


Figure 1.2: The graph drawn by Euler for the Seven Bridges of Königsberg problem.

Through his analysis, Euler discovered a key insight: for a walk to cross each bridge exactly once and return to the starting point, each landmass must be connected to an even number of bridges. In the case of Königsberg, however, every landmass was connected to an odd number of bridges, making such a walk impossible. Euler's solution, published in 1736, was groundbreaking. It not only answered the Königsberg puzzle but also laid the groundwork for an entirely new field of mathematics: *graph theory*. This area of study has since become fundamental to understanding networks, from transportation systems to social media, and even the internet itself. Thus, from a simple riddle about bridges in a small Prussian city, a new mathematical discipline was born—one that continues to influence the world to this day.

Definition 1.1: Graph

A **graph** is a mathematical structure $G = (V, E)$, where V is the set of vertices (or nodes) of G and E is the set of edges that link the vertices of G.

A graph can be **directed** or **undirected**. In a directed graph the edges are *oriented*, meaning that there is difference between the edge (u, v) – going from u to v – and the edge (v, u) – going from v to u. Formally, we have that:

$$E(G) \subseteq V \times V\{(u, v) \mid u, v \in V(G)\}$$

In an undirected graph, instead, the edges are *not oriented*, meaning that there not is difference between the edges (u, v) and (v, u). Formally, we have that:

$$E(G) \subseteq \binom{V(G)}{2} = \{\{u, v\} \mid u, v \in V(G)\}$$

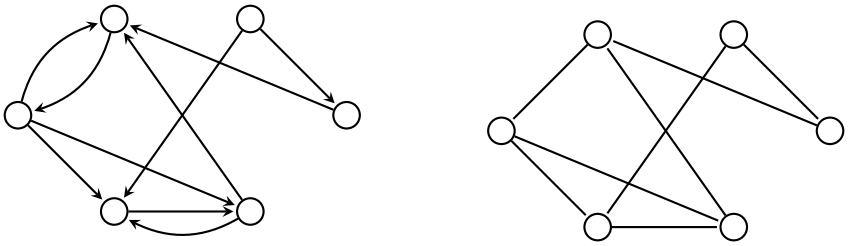


Figure 1.3: A directed graph (left) and an undirected graph (right)

We observe that the definition of graph that we just gave doesn't allow *multiple edges* between two vertices and *loops*, i.e. edges out-going frm and in-going to the same vertex. When this is the case, we say that the graph is **simple**. Generally, simple graphs are enough for any model. Sometimes, however, multiple edges and loops are needed – such as in the Seven Bridges of Königsberg problem. Graphs that allow such edges are called **multigraphs**.

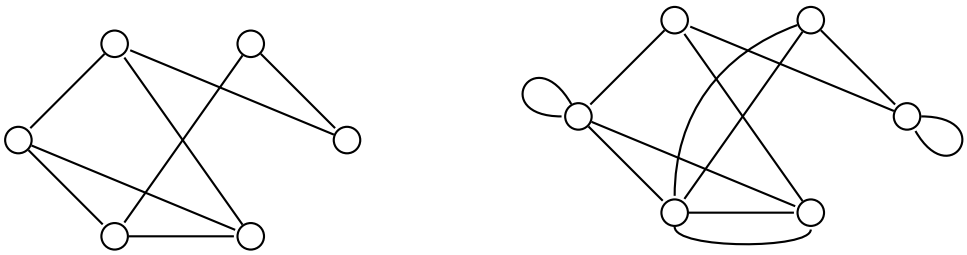


Figure 1.4: A simple graph (left) and a multigraph (right)

From now on, unless stated differently, we'll assume that each graph is simple and undirected. Moreover, to make notation lighter, we will always assume that $|V(G)| = n, |E(G)| = m$ and that $xy = \{x, y\}$ (or $xy = (x, y)$ for directed graphs).

Definition 1.2: Subgraph

Let G be a graph. If H is a graph such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ then H is a **subgraph** of G, written as $H \subseteq G$. A subgraph $H \subseteq G$ is said to be **induced** when for all edges $xy \in E(G)$ such that $u, v \in V(H)$ it holds that $xy \in E(H)$.

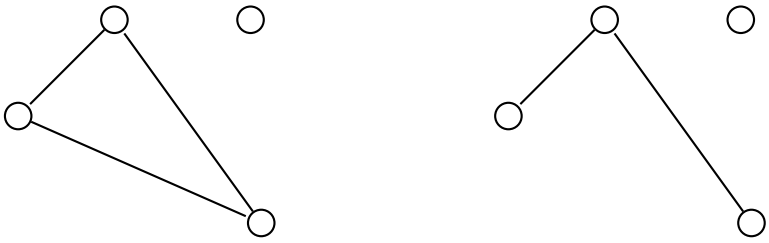


Figure 1.5: Both graphs are a subgraph of the simple graph shown on the left in Figure 1.4. The left subgraph is induced, while the right one is not.

We observe that, by definition, if a subgraph $H \subseteq G$ is induced then H is the unique induced subgraph for the vertices $V(H)$. Hence, when talking about an induced graphs we can consider his set of vertices. Given a subset of vertices $X \subseteq V(G)$, we denote with $G[X]$ the unique induced subgraph of G such that $V(G[X]) = X$, where $E(G[X]) = \{xy \in E(G) \mid x, y \in X\}$.

Definition 1.3: Adjancency, neighborhood and independence

Given a graph G and two nodes $x, y \in V(G)$, we say that x, y are **adjacent** to each other, written as $x \sim y$, if $xy \in E(G)$. For any edge $xy \in E(G)$, we say that xy is incident to x and y. The set of all vertices adjacent to x in G is called **neighborhood**, written as $N_G(x)$, is defined as:

$$N_G(x) = \{y \in V(G) \mid x \sim y\}$$

A subset of vertices X such that for each $x, y \in X$ it holds that $x \not\sim y$ is called **independent set**.

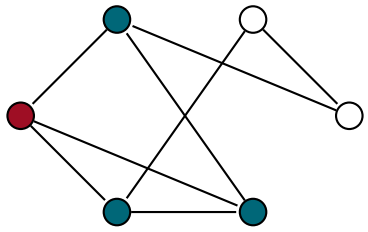


Figure 1.6: The blue nodes form the neighborhood of the red node. The red node and the white nodes form an independent set.

Definition 1.4: Degree

Let G be a graph. Given a vertex $x \in V(G)$, the **degree** of x over G is defined as $\deg_G(x) = |N_G(x)|$. The minimum and maximum degree of G are respectively denoted as $\delta(G)$ and $\Delta(G)$.

$$\delta(G) = \min_{x \in V(G)} \deg_G(x)$$

$$\Delta(G) = \max_{x \in V(G)} \deg_G(x)$$

We say that a graph is **k-regular** if $\delta = \Delta = k$, i.e. every node has degree k

When the context makes it clear, we'll simply write $\deg(x)$ instead of $\deg_G(x)$. The minimum and maximum degree of a graph are two very powerful theorem-proving tools: large portion of results are proven by reasoning on the degree of each node, often proving that some condition does or does not hold. The most basic result involving the degree of a graph is known as the **Handshaking lemma**, which can be stated in two equivalent forms.

Lemma 1.1: Handshaking lemma

For every graph G it holds that:

$$\sum_{x \in V(G)} \deg(x) = 2m$$

Equivalently, for every graph G the number of odd-degree vertices is even.

Proof. It's easy to see that every edge $xy \in E(G)$ is counted exactly two times through $\deg(x)$ and $\deg(y)$, implying that:

$$\sum_{x \in V(G)} \deg(x) = 2m$$

Consider now the subset $X \subseteq V(G)$ containing all the vertices of even degree. We observe that:

$$2m = \sum_{x \in V(G)} \deg(x) = \sum_{x \in X} \deg(x) + \sum_{x' \in V(G)-X} \deg(x')$$

Let $a = \sum_{x \in X} \deg(x)$ and $b = \sum_{x' \in X} \deg(x')$. Since the degrees of the vertices in X are even, we know that a is even. Hence, in order for $2m = a + b$ to hold, b must also be even. However, since each degree in $V(G) - X$ is odd, in order for b to be even it must hold that $|V(G) - X|$ is even. □

1.2 Paths, walks, cycles and trees

After discussing the more general concept of subgraph, we can now focus on particular types of structures that can be usually found in graphs. These sub-structures are the real fundamental tool of reasoning for graph properties.

Definition 1.5: Path

A **path** is a graph P such that $V(P) = \{x_1, \ldots, x_n\}$ and $E(G) = \{x_0x_1, x_1x_2, \ldots, x_{n-1}x_n\}$. The length of a path is defined as the number of edges that form it. A path with n vertices is denoted as P_n.

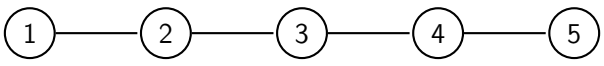


Figure 1.7: The path P_5.

Definition 1.6: Walk

Let G be a graph. A **walk** on G is defined as a sequence $x_0e_1x_1e_2\dots e_{k-1}e_kx_k$ where $x_0,\dots,x_k \in V(G)$ and $e_1,\dots,e_k \in E(G)$. The length of a walk is defined as the number of edges that form it.

When the first and last vertices are equal, i.e. $x_0 = x_k$, we say that the walk is **closed**. We observe that, by definition, a walk allows edges and vertices to be repeated in the sequence. When no vertices in a walk are repeated, the walk corresponds to a path.

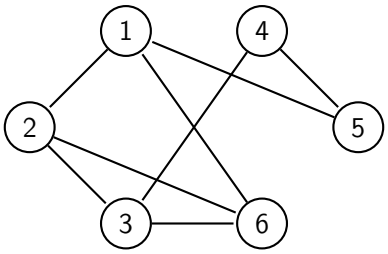


Figure 1.8: The sequence $1\{1,2\}2\{2,6\}6\{6,1\}1\{1,5\}5$ forms a walk on G, but not a path. The sequence $1\{1,2\}2\{2,6\}6\{6,3\}3\{3,4\}4$, instead, forms a path on G.

Proposition 1.1: Paths and walks

Let G be a graph. Given the vertices $x, y \in V(G)$, there is a path from x to y if and only if there is a walk from x to y.

Proof. Since every path is also a walk, the first direction is trivial. Let $W = x_0e_1x_1e_2\dots e_{k-1}e_kx_k$ be the shortest walk. i.e the one with minimum length, such that $x = x_0$ and $y = x_k$. By way of contradiction, suppose that W is not a path. Hence, at least one edge in W must repeat at least once. Let $i, j \in [k]$ be two indices such that $e_i = e_j$. Then, the following sequence W' is a walk from x to y with fewer edges than W, raising a contradiction. Hence, W must be a path.

$$W' = x_0e_1x_1e_2\dots e_ix_ie_{j+1}x_{j+1}e_{j+2}\dots e_{k-1}e_kx_k$$

□

Proposition 1.2

The longest path in any graph has length at least δ.

Proof. If $\delta = 1$ then the longest path is trivially made by one single edge. Suppose now that $\delta \geq 2$, implying that there are at least two vertices in G. Let P be the longest path in G and let $x_1,\dots,x_k$ be its vertices.

Claim: $N(x_k) \subseteq \{x_1,\dots,x_{k-1}\}$.

Proof. By way of contradiction, suppose that there is a vertex $x' \in N(x_k)$ such that $x' \notin \{x_1, \ldots, x_{k-1}\}$. Then, since $x_k \sim x'$, there must be an edge $x_k x'$, implying that the path $P \cup x_k x'$ is longer than P, raising a contradiction. □

Through the claim we easily conclude that $\delta \leq |N(x_k)| \leq k$, meaning that P has length at least δ. □

Definition 1.7: Cycle

A cycle is a graph C such that $V(C) = \{x_1, \ldots, x_n\}$ and $E(G) = \{x_0x_1, x_1x_2, \ldots, x_{n-1}x_n, x_nx_1\}$. The length of a cycle is defined as the number of edges that form it. A cycle with n vertices is denoted as C_n.

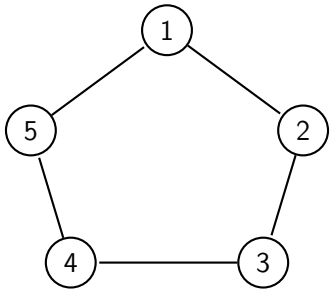


Figure 1.9: The cycle C_5.

Proposition 1.3

In any graph such that $\delta \geq 2$ there is a cycle of length at least $\delta + 1$.

Proof. Let P be the longest path in G and let $x_1, \ldots, x_k$ be its vertices. Through an argument equal to the claim of Proposition 1.2, we know that $N(x_k) \subseteq \{x_1, \ldots, x_{k-1}\}$. Let $i \in [k-1]$ be the minimal index such that $x_i \in N(x_k)$. Since every neighbor of x_k must be inside P and the graph is simple, it must hold that $i \geq \delta$, meaning that the vertices $x_i, x_{i+1}, \ldots, x_{k-1}, x_k, x_i$ form the cycle C_{i+1}. □

Definition 1.8: Connectivity, components and distance

Let G be a graph. Two nodes $x, y \in V(G)$ are said to be linked if there is a path from x to y. If every pair of nodes in G is linked, we say that G is **connected**. If a connected subgraph H of G is maximal – meaning that no other edges can be added to it while preserving connectivity – H is called **component** of G. The **distance** $\text{dist}_G(x, y)$ between two nodes x, y is the length of the shortest path connecting them.

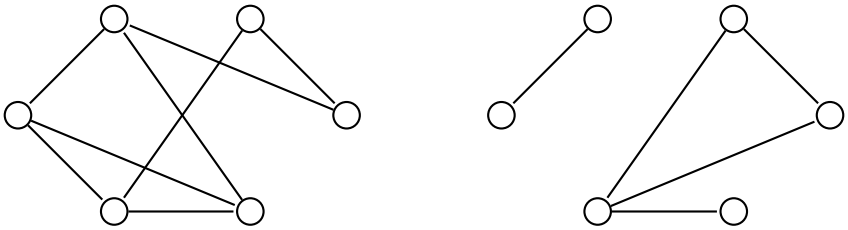


Figure 1.10: A connected graph (left) and a disconnected graph (right). The connected graph has an unique component, while the disconnected graph has two components.

Lemma 1.2

Let G be a graph. If G is connected and C is a cycle in G then for all $e \in E(C)$ it holds that $G - \{e\}$ is still connected.

Proof. Fix an edge $e \in E(C)$. Let $P = x_0 e_1 x_1 \dots e_k x_k$ be a path in G from x to y. If $e \notin E(P)$ then P is also a path in $G - \{e\}$, preserving the connectivity of x and y. Suppose now that $e \in E(P)$. Given $C = z_0 f_1 z_1 \dots f_\ell z_\ell$, without loss of generality assume that $e = e_i = f_1$. Then, the following sequence W is a walk from x to y in $G - \{e\}$ – we cannot be sure that W is a path since P may intersect C on multiple edges.

$$W = x_0 e_1 x_1 \dots x_i f_\ell z_{\ell-1} \dots f_2 z_2 e_{i+1} x_{i+1} \dots e_k x_k$$

By Proposition 1.1, we know that since W is a walk from x to y in $G - \{e\}$ there must also be a path from x to y in $G - \{e\}$, preserving connectivity. □

Definition 1.9: Tree

A **tree** is an connected acyclic subgraph. Any vertex in a tree with degree 1 is called leaf. A rooted tree is a tree with a chosen node called **root**. If every component of a graph is a tree, the graph is referred to as a **forest**.

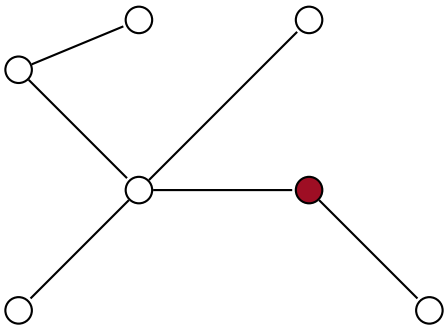


Figure 1.11: A rooted tree. The red node has been chosen as the root.

When the tree is rooted in $r \in V(T)$, we often use the concept of **ancestor** and **parent**. Given two nodes $a, x \in V(T)$, we say that a is an ancestor of x if x lies on a path from r

to x. If p is an ancestor of x and $px \in E(T)$, we say that p is the parent of x. The **least common ancestor (LCA)** between two vertices $x, y \in V(T)$ is the ancestor $z \in V(T)$ shared by x and y that minimizes the value of $\text{dist}(r, z)$. We observe that every pair of vertices of a tree must have a LCA since the root is an ancestor of every node.

Theorem 1.1: Equivalent definitions of tree

Given a graph T, the following statements are equivalent:

1. T is a tree
2. Every vertex pair of T is connected by an unique path
3. T is minimally connected
4. T is maximally acyclic

Proof. We'll proceed by proving a chain of implications.

$1 \implies 2$ Without loss of generality, assume that every pair of nodes has at least one path, since otherwise T is not connected, hence not a tree. Suppose now that there are two vertices $x, y \in V(T)$ that have at least two different paths P, Q from x to y. While traversing P from x to y, let z be the first node such that $z \in V(P) \cap V(Q)$. Similarly, let $w \in V(P) \cap V(Q)$ be the first node encountered while traversing Q from y to x. We observe that since $x, y \in V(P) \cap V(Q)$, the vertices z, w always exist. Given the subpath $P' \subseteq P$ from w to z and the subpath $Q' \subseteq Q$ from z to w, the graph $Q' \cup P'$ is a cycle in T, meaning that T cannot be a tree. By contrapositive, we get that if T is a tree then every pair of vertices is connected by an unique path.

$2 \implies 3$ Suppose that every vertex pair of T is connected by an unique path. Then, T is clearly connected. By way of contradiction, suppose that there is an edge $e \in E(G)$ such that $T - \{e\}$ is still connected. Then, the edge e must be part of at least one of the unique paths connecting two nodes, meaning that such path cannot exist inside $T - \{e\}$, implying that it is not connected. Hence, T must be minimally connected.

$3 \implies 4$ Suppose that T is minimally connected but not maximally acyclic. By way of contradiction, suppose that T has a cycle. Then, by Lemma 1.2 we know that we can remove an edge of the cycle from T and keep it connected, contradicting the fact that T is minimally connected. Hence, T must be acyclic. Pick two vertices $x, y \in V(T)$. Since T is connected, we know that there is a path P connecting x to y. Then, if we were to add the edge xy, the subgraph $P \cup \{xy\}$ would be a cycle in $T \cup \{xy\}$. Thus, T is maximally acyclic.

$4 \implies 1$ Suppose that T is maximally acyclic. Fix a pair of vertices $x, y \in V(T)$. Since T is maximally acyclic, we know that adding the edge xy makes $T \cup \{xy\}$ cyclic. Let C be the cycle in $T \cup \{xy\}$ containing xy. Then, $C - \{xy\}$ must be a path in T from x to y.

□

Lemma 1.3

Let T be a tree. Then:

1. If T has at least two nodes then T has at least a leaf.
2. If $x \in V(T)$ is a leaf then $T - \{x\}$ is still a tree

Proof.

1. By way of contradiction, suppose that T is a tree without leaves. Then, we have that $\delta \geq 2$. However, by Proposition 1.3, in T there must be a cycle with length at least $\delta + 1$, contradicting the very definition of tree. Hence, T must have at least a leaf.
2. Let x be a leaf of T. Since T is acyclic, $T - \{x\}$ is also clearly acyclic. By way of contradiction, suppose that $T - \{x\}$ is not connected. Then, there are at least two vertices $u, v \in V(T - \{x\})$ for which there is a path P between them in T but not in $T - \{x\}$. Since by removing x the vertices u, v became disconnected, the vertex x must lie on P. Moreover, since $u, v \neq x$, the vertex x must be an internal node of the path, meaning that it must have degree 2, contradicting the very definition of leaf.

□

Definition 1.10: Spanning tree

Given a graph G, we say that $T \subseteq G$ is a **spanning tree** of G if T is a tree and $V(T) = V(G)$.

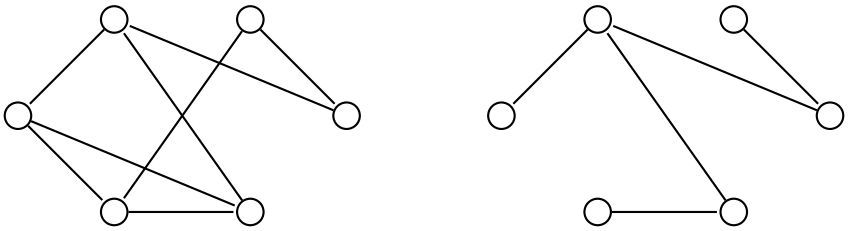


Figure 1.12: A graph (left) and one of its spanning trees (right)

Lemma 1.4

Every connected graph has a spanning tree.

Proof. If G is a tree then clearly it is its own spanning tree. Suppose now that G is a connected graph that is not a tree, meaning that it is acyclic. Then, through Proposition 1.3, we can keep removing the edges $e_1, \ldots, e_k$ from every cycle of G until we reach a minimally connected subgraph T such that $V(T) = V(G)$. By Theorem 1.1, we know that T must be a tree. □

Theorem 1.2

Let T be a connected graph. Then, T is a tree if and only if it has $n-1$ edges.

Proof. We proceed by induction on n. If $n = 1$ then T is trivially the tree with 0 edges. If $n > 1$, instead, through the previous lemma we know that T must have at least a leaf $x \in V(T)$ for which $T - \{x\}$ is still a tree. By inductive hypothesis, we know that $T - \{x\}$ has $n-2$ edges. Hence, by adding the unique edge incident to x, we get that T has $n-1$ edges.

Vice versa, by way of contradiction, suppose that T is connected, has $n-1$ edges but that it is not acyclic. Then, through the previous lemma we know that T must have a spanning tree T'. Moreover, since T has a cycle and T' does not, we know that T' must have fewer edges than T. However, we have just proven that every tree must have $n-1$ edges. Hence, we get that $n - 1 = |E(T')| < |E(T)| = n - 1$, which is a contradiction. Hence, T must also be acyclic. □

1.3 Complete graphs and bipartite graphs

Definition 1.11: Complete graph

A **complete graph** is a graph where every pair of vertices is adjacent to each other. A complete graph with n vertices in denoted as K_n. An induced subgraph that is complete is called **clique**.

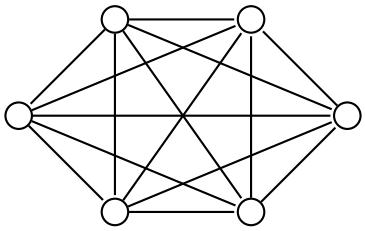


Figure 1.13: The complete graph K_6

Definition 1.12: Bipartite graph

A **bipartite graph** is a graph with a subset of vertices $X \subseteq V(G)$ such that for every edge $\{u, v\} \in E(G)$ it holds that $u \in X$ and $v \in V(G) - X$. Equivalently, we have that both X and $V(G) - X$ are independent sets. The pair $(X, V(G) - X)$ is called **bipartition** of the graph.

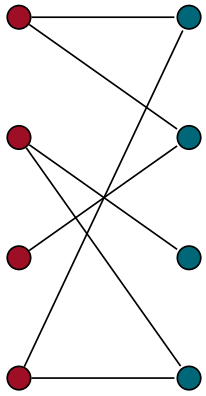


Figure 1.14: The red and blue nodes form a bipartition of the graph.

It's easy to see that no complete graph can be bipartitioned since it's impossible to separates the nodes into two independent sets. In fact, by definition, cliques are the very opposite of the concept of independent set. Through the following lemma we can extend the claim on complete graphs to cliques: if G contains a clique of any cardinality then G cannot be bipartitioned.

Proposition 1.4

G is bipartite if and only if every subgraph of G is bipartite

Proof. The converse implication trivially holds. Let H be a subgraph of G. Then, any bipartition $(X, V(G) - X)$ of G induces a bipartition $(X \cap V(H), V(H) - X)$ over H. □

Lemma 1.5

G is bipartite if and only if every component of G is bipartite

Proof. The direct implication trivially follows from the previous proposition. Given the components $H_1, \dots, H_k$ of G, let $(X_1, V(H_1) - X_1), \dots, (X_k, V(H_k) - X_k)$ be the bipartitions of the components. Then, since each component is by definition disjoint from the others, the pair $(X, V(G) - X)$ where $X = \bigcup_{i \in [k]} X_i$ is a bipartition of G. □

We observe that the above characterizations of bipartite graphs are useful but still "weak". A stronger characterization can be achieved through the following theorem. Given a path P and two vertices $x, y \in V(P)$, we denote the subpath of P from x to y with xPy.

Theorem 1.3: Equivalent defintion of bipartite graphs

G is bipartite if and only if it doesn't contain odd cycles.

Proof. Through the above proposition, it's sufficient to prove that for all $k \in \mathbb{N}$, C_{2k+1} is not bipartite. Fix $k \in \mathbb{N}$. By way of contradiction, suppose that there is bipartition

$(X, V(G) - X)$ of C_{2k+1}. Let $x_1, \ldots, x_{2k+1}x_1$ be the edges of C_{2k+1}. Without loss of generality, assume that $x_1 \in X, x_2 \notin X, x_3 \in X, \ldots, x_{2k} \notin X$. Then, if $x_{2k+1} \in X$ then $x_{2k+1}x_1 \subseteq X$, while if $x_{2k+1} \notin X$ then $x_{2k}x_{2k+1} \subseteq V(G) - X$. In both cases, we get that the pair cannot be a bipartition.

Vice versa, suppose that G is not bipartite. Then, through the previous lemma at least one component H of G is not bipartite. Let T be a spanning tree of H and fix $r \in V(T)$ as its root. Let $X = \{x \in V(T) \mid \text{dist}_T(r, x) \text{ is even}\}$. By definition, $(X, V(T) - X)$ for a bipartition of T. Hence, since T is bipartite but H isn't, there must be an edge $xy \in E(H)$ such that either $x, y \in X$ or $x, y \notin X$. Let P_x and P_y respectively be the paths in T from x to r and from y to r. Let z be the LCA of x and y in T.

Claim: $C = zP_xx \cup zP_yy \cup xy$ is an odd cycle

Proof of the claim. Since either $x, y \in X$ or $x, y \notin X$ holds, we know that $\text{dist}(r, x)$ and $\text{dist}(r, y)$ must be either both even or both odd. Hence, the lengths of P_x and P_y must share the same parity. Moreover, since z is the LCA of x and y w have that $rP_xz = rP_yz$. Thus, the lengths of zP_xx and zP_yy must also share the same parity. This concludes that $zP_xx \cup zP_yy \cup xy$ must be an odd cycle. □

Since C is an odd cycle and it is a subgraph of T, and thus of G, we conclude that G contains an odd cycle. □

1.4 Eulerian tours and Hamiltonian paths

At the start of this chapter, we introduced the Seven Bridges of Königsberg problem, which led to the emergence of graph theory as a branch of combinatorics. In modern graph theory, the problem is formalized through the concept of *Eulerian tour*.

Definition 1.13: Eulerian tour

An Eulerian tour over a graph G is closed walk that traverses every edge of G exactly once.

To solve the problem, Euler [Eul41] proved the following theorem, which implies that the answer is "no" since the multigraph that models the problem contains some odd-degree vertices.

Theorem 1.4: Euler's theorem

A graph (or multigraph) has an Eulerian tour if and only if G is connected and every vertex has even degree

Proof. By way of contradiction, suppose that G has an Eulerian tour W. By way of contradiction, suppose that G is not connected. Then, we get an easy contradiction: if G is disconnected then W cannot traverse every edge of the graph. Hence, G must be connected. Again, by way of contradiction suppose that G has at least one odd-degree

vertex $x \in V(G)$. Let $\deg(x) = 2k+1$. Since W is a closed walk, we can assume without loss of generality that x is the first vertex of the walk. When traversing W starting from x, one of the edges incident to x is crossed, hence we have $2k$ incident edges left. Every time the tour returns to x, two edges are crossed – one in-going and one out-going. Hence, in order to cover all the edges, the tour has to return to x for k times. However, this implies that there are no more edges left to close the tour on x, raising a contradiction. Hence, the vertex x cannot exist.

Vice versa, assume that G is connected and every vertex has even degree. Let $W = x_0 e_1 x_1 \dots e_k x_k$ be the longest walk over G with no repeating edges.

Claim 1: $x_k = x_0$, i.e. W is a closed walk

Proof of Claim 1. By way of contradiction, suppose that $x_k \neq x_0$. Let $2\ell + 1$ be the number of edges incident to x_k in W – one edge is given by e_k while 2ℓ edges are given by edges needed to cross the other ℓ the vertices $x_{i_1}, \dots, x_{1_\ell}$ such that $x_i = x_k$. Since x_k has even degree, we know that there must be another edge $\{x_k, y\} \in E(G - W)$, implying that $W \cup \{x_k, y\}$ is walk longer than W, which is absurd. □

Claim 2: W contains every edge of G

Proof of Claim 2. By way of contradiction, suppose that that there is an edge $uv \in E(G - W)$. Fix $i \in [k]$. By connectivity of G, we know that there must be a path P disjoint from x_i to u or from x_i to v. Without loss of generality, assume that P is a path from x_i to u. Since $uv \notin E(W)$, there must be an edge $x_j y \in E(P - W)$ outgoing from W. Since W is a closed walk, we can assume without loss of generality that x_j is the first (and last) vertex of the walk. Then, the walk $W \cup x_j y$ is walk that doesn't repeat any vertices longer than W, raising a contradiction. □

Through the two claims, we conclude that W is an Eulerian tour. □

Another very interesting concept similar to Eulerian tours was formulated by Hamilton: closed walks that touch every node exactly once instead of every edge exactly once. We observe that this condition implies that the closed walk is nothing more than the subgraph P_n. Surprisingly, finding Hamiltonian path inside a graph is way harder than finding an Eulerian tour: the former is an NP-hard problem, while the latter can be solved in polynomial time. A variant of this concept is the Hamiltonian cycle, i.e. a cycle that goes through all the nodes of a graph

Definition 1.14: Hamiltonian paths and cycles

An Hamiltonian path over a graph G is a subgraph $P_n \subseteq G$. An Hamiltonian cycle over a graph G is a subgraph $C_n \subseteq G$.

When some conditions are met, an Hamiltonian cycle (hence also an Hamiltonian path) is guaranteed to exist. For instance, Dirac [Dir52] formulated the following theorem.

Theorem 1.5: Dirac's theorem

Let G be a graph. If $\delta \geq \frac{n}{2}$ then there is an Hamiltonian cycle in G.

Proof. First of all, we show that Dirac's condition forces the graph to be connected.

Claim 1: G is connected

Proof of Claim 1. By way of contradiction, suppose that G is not connected. Then, G has at least two components. Hence, the smallest component H of G can have at most $\frac{n}{2}$ edge. However, since $\delta \geq \frac{n}{2}$, each node $x \in H$ must have at least $\frac{n}{2}$ neighbors. Thus, since $\{x\} \cup N(x) \subseteq H$, we get that H must have at least $\frac{n}{2}+1$ nodes, raising a contradiction. □

Let P be the longest path on G an let $x_0, \ldots, x_k$ be its vertices.

Claim 2: There is an index ℓ such that $x_0, \ldots, x_k, x_{\ell-1}, x_\ell, x_0$ is a cycle

Proof of Claim 2. Using the same argument as Proposition 1.2, we know that $N(x_0), N(x_k) \subseteq \{x_0, \ldots, x_k\}$. Let I_0 and I_k be defined as:

$$I_0 = \{i \mid 1 \leq i \leq k, x_i \in N(x_0)\} \qquad I_k = \{i \mid 1 \leq i \leq k, x_{i-1} \in N(k)\}$$

Since $\delta \geq \frac{n}{2}$, we know that $\frac{n}{2} \leq |I_0|, |I_k| \leq n-1$, by the pigeonhole principle there must be at least one index $\ell \in I_0 \cap I_k$, meaning that $x_0 \sim x_\ell$ and $x_k \sim x_{\ell-1}$. Thus, the sequence of nodes $x_0, \ldots, x_k, x_{\ell-1}, x_\ell, x_0$ is a cycle. □

Let C be the cycle given by $x_0, \ldots, x_k, x_{\ell-1}, x_\ell, x_0$. By way of contradiction, suppose that C has less than n nodes. Given $y \in V(G-C)$, by connectivity of G there must be an index i for which there is a path P connecting x_i and y. However, this implies that the following path

$$P \cup x_i x_{i+1} \cup \ldots \cup x_k x_0 \cup \ldots \cup x_{i-2} x_{i-1}$$

is longer than P, raising a contradiction. Hence, C must be an Hamiltonian path. □

We observe that Dirac's condition, i.e. $\delta \geq \frac{n}{2}$ is actually optimal, meaning that we cannot find a better lower bound that guarantees the existence of an Hamiltonian cycle. For instance, there are many graphs with $\delta = \frac{n}{2} - 1$ for which there is no Hamiltonian cycle.

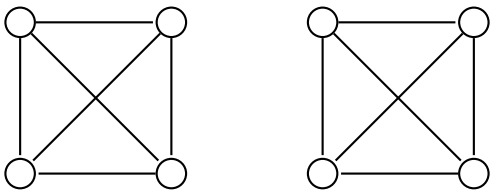


Figure 1.15: Example of a graph with $\delta = \frac{n}{2} - 1$ without Hamiltonian cycles.

1.5 Solved exercises

Problem 1.1

For any graph G, prove that at least two vertices have the same degree.

Proof. By way of contradiction, suppose that no vertices share the same degree. Then, the function $\deg : V \to \{0, \ldots, n-1\}$ must be bijective, implying that there are two vertices $u, v \in V$ such that $\deg(u) = 0$ and $\deg(v) = n - 1$, which is absurd since there should both be and not be an edge between u and v. Hence, at least two vertices must share have the same degree. □

2

Graph matchings

2.1 Maximum matching

In graph theory, a **matching** in a graph is a set of edges that do not have a set of common vertices. In other words, a matching is a graph where each node has either zero or one edge incident to it. Graph matching has applications in flow networks, scheduling and planning, modeling bonds in chemistry, graph coloring, the stable marriage problem, neural networks in artificial intelligence and more.

Definition 2.1: Matching

Given a graph G, a matching over G is a subset $M \subseteq E(G)$ such that $\forall e, e' \in M$ it holds that $e \cap e' = \varnothing$

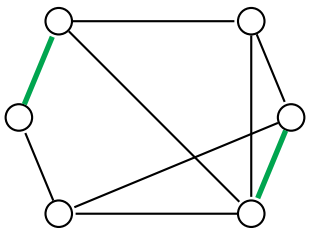


Figure 2.1: The green edges form a matching of the graph.

We're often interested in finding the matching with maximum cardinality. Before proceeding, it's important to distinguish between the concepts of **maximal** and **maximum**. In general, given a property P, a sub-structure X of a structure S is said to be maximal for P over S if $P(X)$ is true and there is no other sub-structure X' of S such that $P(X')$ is true and X is contained inside X'. Instead, X is said to be the maximum for P over S if $P(X)$ is true and there is no other sub-structure X' of S with a higher value for the property $P(X)$. For instance, the matching shown in the above figure is maximal because

it cannot be extended with other edges without breaking the matching property, but it's not a maximum matching.

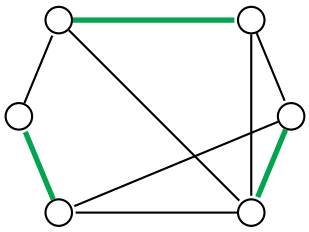


Figure 2.2: Maximum matching for the previous graph.

In particular, we'll focus on matching on **bipartite graphs**. These types of matchings are of particular interest due to how they describe many real-life situations. For instance, the problem of finding the optimal assignment of tasks to a group of employees can be solved by finding a maximum matching: we split the graph in two partitions, one with all the tasks and one with all the employees, connecting each employee to the tasks that he's capable of completing.

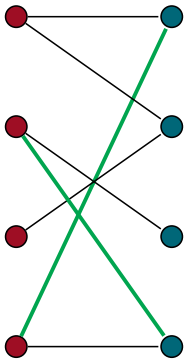


Figure 2.3: A matching on a bipartite graph.

Consider the matching on the bipartite graph shown above. We observe that this matching is neither maximal nor maximum. When the matching is not maximal, we can obtain a matching with greater cardinality simply by extending it.

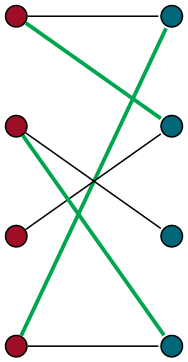


Figure 2.4: A maximal matching on a bipartite graph.

We notice that the above matching is maximal, but not maximum. For instance, we observe that the number of edges outside of the matching that we selected form new matching with more edges than the one that we have considered. Hence, by *swapping* all the edges, we get a matching with higher cardinality. Moreover, this new matching is clearly a maximum one since the number of nodes is equal to twice the number of selected edges.

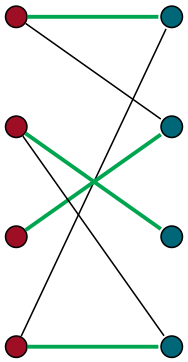


Figure 2.5: The maximum matching obtained by swapping all the edges.

Let's take a closer look to what we just did. We notice that all the edges of the last non-maximum matching actually form a path whose edges alternate between being outside of the matching and inside of the matching. Moreover, both the first and last edge of such path are outside of the matching, hence the number of outside edges is one more than the number of inside edges. We generalize such paths through the concept of *alternating path* and *augmenting path*.

Definition 2.2: Alternating and augmenting path

Let G be a bipartite graph and let $M \subseteq E(G)$ be a matching on G. An **M-alternating path** is a path starting from an unmatched node and whose edges alternate between M and $E(G) - M$. If the path also ends at an unmatched node, the path is said to be **M-augmenting**.

To clarify this definition, let's make things more formal. A path $P = x_0 e_1 x_1 e_2 \dots e_k x_k$ is M-alternating when x_0 is unmatched, meaning that there is no edge $e \in M$ in the matching such that $x_0 \in e$, and whose edges alternate between being outside of the matching and inside of the matching, meaning that $e_1 \notin M, e_2 \in M, e_3 \notin M$ and so on. When x_k is also unmatched, the path is said to be M-augmenting. This name comes from the fact that, in order for x_k to be unmatched, the last edge is not inside the matching, meaning that we have more edges outside of the matching then inside of it. Moreover, since x_0 and x_k are unmatched, by swapping the edges inside of the matching with the edges outside of it we're guaranteed to get a matching (with more edges than the previous one).

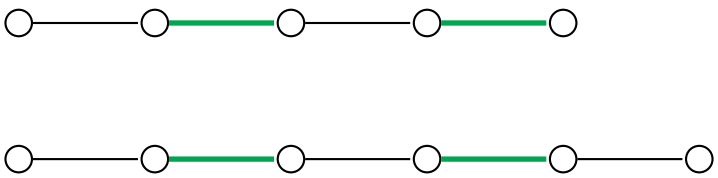


Figure 2.6: An M-alternating path (above) and an M-augmenting path (below).

Lemma 2.1

Let G be a bipartite graph and let $M \subseteq E(G)$ be a matching on G. If P is an M-augmenting path then $M \Delta E(P)$ is matching on G with more edges than M

Proof. First, we recall that the symmetric difference $M \Delta E(P)$ is defined as follows:

$$M \Delta E(P) = (M \cup E(P)) - (M \cap E(P))$$

Let $M' = M \Delta E(P)$. We observe that all the edges that aren't shared by M and P are also not in M', hence we can ignore them. Let $P = x_0 e_1 x_1 e_2 \dots e_k x_k$. Since P is augmenting, the number of edges in P that are also in M is less than the number of edges that aren't in M. Hence, we get that $|M'| > |M|$. □

The above lemme gives us an easy way to increase the cardinality of our matching by finding augmenting paths and swapping edges. But how can we be sure that we have reached the maximum matching? Berge [Ber57] proved that the above lemma can indeed be extended: if there are no augmenting paths then the matching is maximum.

Theorem 2.1: Berge's theorem

Let G be a bipartite graph and let $M \subseteq E(G)$ be a matching on G. Then, M is a maximum matching on G if and only if there are no M-augmenting paths.

Proof. One of the two implications directly comes from the previous lemma. For the other implication, we prove the contrapositive. Suppose that M is a non-maximum matching on G. Then, there is at least one matching M' on G such that $|M'| > |M|$. Let $H \subseteq G$ be the graph such that $V(H) = V(G)$ and $E(H) = M \Delta M'$, where Δ denotes the *symmetric difference*. We observe that every edge that is shared among M and M' gets deleted in H, hence we can ignore them.

Claim: for every $x \in V(H)$ it holds that $\deg_H(x) \leq 2$.

Proof of Claim 1. Since M and M' are both matchings on G, each vertex can have at most one edge in M and at most one edge in M'. If they share the same edge then H won't contain such edge. If they don't, x will have degree 2 in H. □

We observe that the above claim has many consequences. In particular, it implies that every component of H must be either a cycle or a path (including trivial paths of one single vertex).

Claim 2: every cycle component of H has even length

Proof of Claim 2. By way of contradiction, suppose that there is a cycle C of odd length. By construction, each component has to alternate between edges of M and M'. Hence, at least one vertex of C must have both edges lying either inside M or M', contradicting the fact that either M or M' is a matching. □

Since $|M'| > |M|$, there must be a component with at least one (and at most one) edge in M'. Since the edges of each component alternate between M and M', by Claim 2 we know that such component cannot be a cycle. Hence, it must be a path component P. In order for P to have more edges in M' than edges in M, the first and last edge must be edges of M', meaning that P is an M-augmenting path. By contrapositive, if there is no M-augmenting path then M is a maximum matching. □

This theorem has been used to construct many algorithms for finding a maximum matching on bipartite graphs. The most famous one is the **Hopcroft-Karp algorithm** [HK73], due to a guaranteed runtime of $O(m\sqrt{n})$. Given a bipartite graph G with bipartition (A, B), the idea behind such algorithms is to run a simultaneous BFS starting from all the unmatched vertices of A, until at least one free node of B is found. Then, we run a DFS over the forest produced by the BFS, starting from the unmatched nodes of B. Each path found through this procedure is an augmenting path, hence their edges can be swapped to increase the cardinality of the matching. The whole process is repeated until no augmenting path is found.

What about non-bipartite graphs? For the general case, problems rise with odd cycles, which never exist in bipartite graphs (Theorem 1.3). Any matching over an odd-length cycle will never cover every vertex of the cycle. This means that in any odd-length cycle there must be a vertex adjacent to two edges which cannot be part of the matching considered. This makes finding the optimal matching hard to find on such graphs. In 1965, Edmonds [Edm65] came up with the **Blossom algorithm**, a real piece of art in the world of algorithm design. The algorithm is based on the repeated contraction and distension of *blossoms*, i.e. a cycles of odd length which contain the maximal number of edges in the current matching. The *blossom lemma* states that any matching found on the graph with every blossom contracted is a maximum matching if and only if it is also a maximum matching for the graph original graph.

The maximum matching problem is highly related to the **Minimum Vertex Cover** problem. This problem involves finding the smallest subset of vertices in a graph such that every edge is incident to at least one vertex in the set.

Definition 2.3: Vertex Cover

Given a graph G, a vertex cover over G is a subset $C \subseteq V(G)$ such that $\forall e \in E(G)$ there is a vertex $v \in C$ such that $v \in e$.

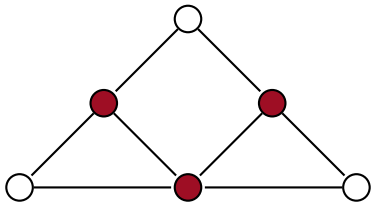


Figure 2.7: The red nodes are the smallest possible vertex cover of the graph.

Proposition 2.1

Given a graph G, for every matching M on G and every vertex cover V of G it holds that $|M| \leq |V|$.

Proof. By definition, we observe that if V is a vertex cover for G then it is also a vertex cover for any subgrpah $G' \subseteq G$. Hence, V is also a vertex cover for any G_M, where $V(G_M) = V(G)$ and $E(G_M) = M$. By definition of matching, in G_M any vertex has either degree 0 or 1. Thus, each vertex of V can cover at most one edge of M, meaning that V has to have at least $|M|$ vertices to cover all the edges of M. □

In bipartite graphs, the above proposition can be strengthened. In 1931, Kőnig [Sza20] proved that the cardinality of the maximum matching and the minimum vertex cover is equal. This implies that the minimum vertex cover can be efficiently found on bipartite graphs by finding the maximum matching.

Theorem 2.2: Kőnig's theorem

Given a bipartite graph G, let M^* and V^* respectively be a maximum matching and a minimum vertex cover on G. Then, it holds that $|M^*| \leq |V^*|$.

Proof. We observe that it suffices to show that there is a (non-minimum) vertex cover V such that $|V| = \mathcal{M}^*$ in order to get $|M^*| \leq |V^*| \leq |V| = |M^*|$. Let (A, B) be the bipartition of G. We define V as the set of vertices such that $\forall ab \in M$, with $a \in A$ and $b \in B$, if there is an alternating path starting from a vertex in A and ending on b then $b \in V$, otherwise $a \in$ U. Fix an edge $xy \in E(G)$ with $x \in A$ and $y \in B$.

Claim: V covers xy

Proof. Suppose that $xy \in M$ then we know that either x or y is in V by definition of U itself, meaning that such edge is covered. Hence, we may assume that $xy \notin M$. We have two cases:

- x is unmatched, meaning that $\nexists uv \in M$ such that $u = x$. Then, y must be matched by M^* to some $u \in A$, since otherwise $M^* \cup \{xy\}$ would be a matching greater than M^*. Hence, the edge uy is an alternating path starting from A and ending on y, meaning that $y \in V$.

- x is matched, meaning that $\exists xv \in M$. Then, by definition of V, either x or v must lie inside V. If $x \in$ U then xy is trivially covered by V. If $v \in V$, instead, by definition of V there must be an alternating path P starting from A and ending at v. This also implies that $P \cup vx \cup xy$ is an alternating path ending on y. However, this implies that y must be matched through some edge $wy \in M$, with $w \neq x$, since otherwise the path $P \cup vx \cup xy$ would be an augmenting path, contradicting the fact that M^* is maximum (Berge's theorem). Thus, we conclude that $y \in V$ since $wy \in M$ and $P \cup vx \cup xy$ is an augmenting path that ends on y.

□

In both cases, we get that either x or y is inside V, concluding that xy is covered by U. Applying the same argument over all edges, we get that V is a vertex cover. □

2.2 Perfect matching

We'll now focus on a particular type of matching on graphs. First, we notice that for any matching M on a graph G (even when G is non-bipartite) it must always hold that:

$$|M| \leq \frac{|V(G)|}{2}$$

When the inequality is satisfied at equality, the matching is said to be **perfect**.We observe that such condition is equivalent to saying that every vertex of the graph is covered by an edge. In fact, the concept of perfect matching can be, in some sense, viewed as the edge-version of a vertex cover, even though there is an additional constraint for the disjointness of the edges.

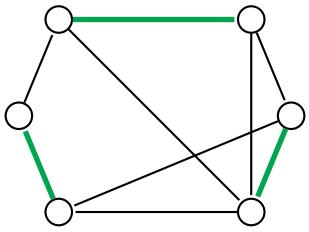


Figure 2.8: A perfect matching.

Definition 2.4: Perfect matching

A matching M on a graph G is said to be perfect if for all vertices of G there is an edge of M covering it.

We observe that every perfect matching is clearly a maximum one, but the contrary doesn't always hold. Moreover, not all graphs can have a perfect matching – trivially, a graph with an isolated vertex cannot have a perfect matching.

For bipartite graphs, the concept of perfect matching has to be slightly tweaked: if (A, B) is the bipartition of the graph and $|A| \neq |B|$, there is no way for a perfect matching to hold. Hence, assuming that $|A| \leq |B|$, we use the concept of **A-perfect matching**, i.e. a matching that covers every vertex of A – notice that if (A, B) is a partition then (B, A) is also a partition, hence we can always assume that $|A| \leq |B|$.

Hall [Hal35] was able to characterize A-perfect matchings on bipartite graphs through the now now called *Hall's Marriage Condition* – the name comes from the original set-theoretic formulation of the problem.

Theorem 2.3: Hall's marriage condition

Let G be a bipartite graph with bipartition (A, B) with $|A| \leq |B|$. Then, G has an A-perfect matching if and only if for all subsets $S \subseteq A$ it holds that $|N(S)| \geq |S|$, where $N(S) = \bigcup_{s \in S} N(s)$.

Proof. Suppose that there is an A-perfect matching M of G. Fix a subset $S \subseteq A$. By definition of M, each node $s \in S$ is matched through M with a different node, therefore $|N(S)| \geq |S|$.

Vice versa, suppose that there is no A-perfect matching M. Let V^* be a minimum vertex cover on G. By Kőnig's theorem, we know that the size of V^* has to be equal to the size of any maximum matching of G. However, since there is no A-perfect matching, any maximum matching has to have size less than $|A|$, meaning that $|V^*| < |A|$. Consider now the set $S = A - V^*$. Since G is bipartite, it must hold that $N(S) = V^* \cap B$. We observe that:

$$|N(S)| \leq |V^* \cap B| = |V^*| - |V^* \cap A|$$

Since $V^* \cap A = A - (A - V^*) = A - S$, we get that:

$$|N(S)| \leq |V^*| - |V^* \cap A| = |V^*| - |A| + |S|$$

Finally, since $|U| < |A|$, we conclude that:

$$|N(S)| \leq |V^*| - |V^* \cap A| = |V^*| - |A| + |S| < |S|$$

□

What if the graph is non-bipartite? Can perfect matchings also be characterized for the general case? The answer is yes! Before proving the result, we'll build an intuition for it. First, we observe that each graph with an odd number of vertices cannot have a perfect matching since there will always be at least one unmatched vertex. This can be extended to components: if a graph contains at least one component with an odd number of vertices then there will always be at least one unmatched vertex inside that component. But what if such components get "connected" through an intermediate vertex x? Then, once such vertex gets matched, the problem still holds: the induced graph $G[V(G) - x]$ now contains components with an odd number of vertices.

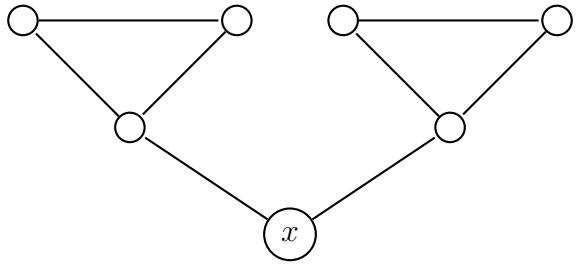


Figure 2.9: Once the vertex x gets matched, the graph $G[V(G) - x]$ is made of two odd components, making a perfect matching impossible.

But what if we add another edge? In that case, we would have enough vertices to complete the perfect matching.

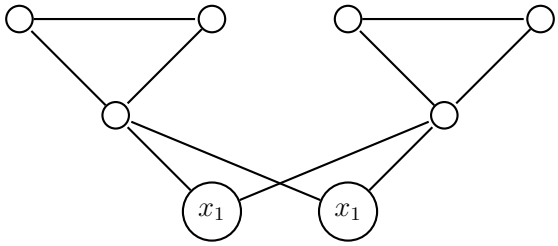


Figure 2.10: Once the vertex x_1 gets matched with an edge touching one of the two odd components, x_2 can be matched through the edges touching the other component.

Thus, it's easy to see that if there is a subset of vertices $S \subseteq V(G)$ such that $\mathcal{O}(G[V(G) - S]) > |S|$ then there cannot be a perfect matching, where $\mathcal{O}(G)$ is the number of odd components of G. To make things easier to read, we say that a graph satisfies the **perfect matching condition** when for all subsets $S \subseteq V(G)$ it holds that $\mathcal{O}(G[V(G) - S]) \leq |S|$. In 1947, Tutte [LP09] proved that such condition perfectly characterizes the existence of a perfect matching. To show the result, we first prove the following lemma.

Lemma 2.2

Let G be a graph and let G' be a super-graph of G, i.e. $G \subseteq G'$, obtained by adding edges to G and keeping the same vertices. Then, if G' doesn't satisfy the perfect matching condition then G also doesn't.

Proof. Suppose that G' doesn't satisfy the condition, meaning that $\exists S \subseteq V(G)$ such that $\mathcal{O}(G[V(G') - S]) > |S|$. Then, since G' is obtained by adding edges, the number of odd components in G' may remain the same of G or decrease if two previously disconnected components are now connected in G', meaning that:

$$|S| < \mathcal{O}(G[V(G') - S]) \leq \mathcal{O}(G[V(G) - S])$$

and thus that G also doesn't satisfy the condition. □

Theorem 2.4: Tutte's theorem

Let G be a graph. Then, G has a perfect matching if and only if for all subsets $S \subseteq V(G)$ it holds that $\mathcal{O}(G[V(G) - S]) \leq |S|$.

Proof. Suppose that there is a perfect matching M on G. Fix a subset $S \subseteq V(G)$. Then, every odd component of $G[V(G) - S]$ sends at least one matching edge to a distinct vertex of S, meaning that $|S| \geq \mathcal{O}(G[V(G) - S])$.

Vice versa, suppose that there is no perfect matching on G. Let G' be the maximal supergraph of G that doesn't have a perfect matching obtained by adding edges and keeping the same vertices. We say that a set X satisfies the *clique-adjacency condition* if every component of $G'[V(G') - X]$ is a clique and every $u \in X$ is adjacent to every $v \notin X$

Claim 1: if there is a subset $S' \subseteq V(G')$ that satisfies the clique-adjacency condition then G doesn't satisfy the perfect matching condition.

Proof of Claim 1. Assume that S' is a subset satisfying the clique-adjacency condition. We observe that, since very component of $G'[(V(G)) - S']$ is a clique, for each even component we can always find a perfect matching restricted to it. For odd components, instead, we can always find a perfect matching over all its nodes except for one. Let $M_1, \ldots, M_k$ be the perfect matching defined on the even components and let $M'_1, \ldots, M'_h$ be the p.m. defined on the odd components.

If S' violates the perfect matching condition on G then we're done. Otherwise, if S' doesn't violate it then we know that $|S'| \geq \mathcal{O}(G[V(G) - S'])$. Hence, since every vertex of S' is adjacent to every vertex that isn't in S', we can match the unmatched vertex of each odd component of $G'[V(G) - S']$ with a vertex of S'. We observe that the remaining vertices of S' must form a clique, since otherwise we could add edges to G' and preserve the absence of a perfect matching (the only way to prevent this is if all the edges are already in the graph). Moreover, this clique must have an odd number of nodes, since otherwise we could form a perfect matching on G'. Hence, G' must have an odd number of total vertices (thus also G does), meaning that the empty set $\varnothing$ violates the perfect matching condition on G. Hence, at least one subset of G must violate the perfect matching condition. □

Let $S = \{v \in V(G') \mid \deg_{G'}(v) = n - 1\}$. By way of contradiction, suppose that that S doesn't satisfy the clique-adjacency condition. Then, there is a component of $G'[V(G') - X]$ that isn't a clique or there is a $u \in S$ and a $v \notin S$ such that $u \not\sim v$. However, by definition of S, the second case cannot hold, hence there must be a component that isn't a clique, meaning that there are at least two vertices x, y in that component such that $x \not\sim y$.

Let P be the shortest path from x to y in $G'[V(G') - S]$. Let a, b, c be the first vertices of P. We observe that $a \not\sim c$ since otherwise P wouldn't be the shortest path. Moreover, since P is a path in $G'[V(G') - S]$, we know that $b \notin S$. Hence, there must be a vertex $d \in V(G)$ such that $b \not\sim d$. In other words, we obtain a kite-shaped subgraph (see Figure 2.11).

By maximality of G', we know that $G' \cup \{ac\}$ must have a perfect matching M_1. Likewise, $G' \cup \{bd\}$ must have a perfect matching M_2. Furthermore, it must hold that $ac \in M_1$ and $bd \in M_2$ since otherwise M_1 and M_2 would be perfect matchings on G'.

Claim 2: $(M_1 - \{ac\}) \cup (M_2 - \{bd\})$ contains a perfect matching on G'

Proof of Claim 2. Let $\overline{M} = (M_1 - \{ac\})\Delta(M_2 - \{bd\})$. Since M_1 and M_2 are two matchings, every vertex of $\overline{M}$ has degree at most 2. Hence, every component of $\overline{M}$ must be either an alternating path or an alternating cycle. Moreover, we observe that the only vertices that have degree 1 are a, b, c, d. Thus, these vertices must be the endpoints of two components P_1, P_2 that are alternating paths. We observe that, since $ac \in M_1$ and $bd \in M_2$, one of such paths must be an M_1-alternating path, while the other must be an M_2-alternating path. Without loss of generality, let P_1 be the M_1-alternating path and let P_2 be the M_2-alternating path. We notice that $M' = (M_1 \cup M_2) - (E(P_1) \cup E(P_2))$ is a perfect matching on $G - (V(P_1) \cup V(P_1))$. We have three cases:

1. P_1 is a path from a to c and P_2 is a path from b to d. Then, $(M_1 \cap E(P_2)) \cup (M_2 \cap E(P_1)) \cup M'$ is a perfect matching on G'
2. P_1 is a path from a to b and P_2 is a path from c to d. Then, $(M_1 \cap E(P_2)) \cup (M_2 \cap E(P_1)) \cup M' \cup \{bd\}$ is a perfect matching on G'
3. P_1 is a path from a to d and P_2 is a path from b to d. Then, $(M_1 \cap E(P_2)) \cup (M_2 \cap E(P_1)) \cup M' \cup \{ac\}$ is a perfect matching on G'

□

Since $(M_1 - \{ac\}) \cup (M_2 - \{bd\})$ always contains a perfect matching on G', we conclude that S it is impossible for S to not satisfy the clique-adjacency condition. Thus, through Claim 1 and the previous lemma we know that both G' and G violate the perfect matching condition. □

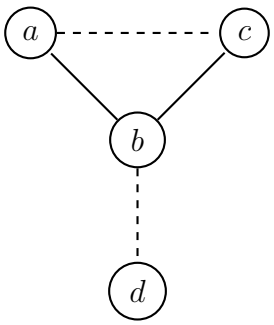


Figure 2.11: The kite-shaped subgraph yield by Tutte's argument. The dashed edges represent non-existing edges.

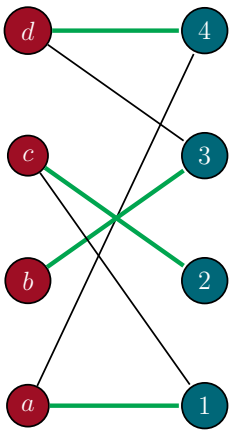
2.3 Stable matching

We'll now focus on a more advanced type of maximum, i.e. the **Stable Marriage** problem. This problem acts as a generalization of the common maximum matching problem, where the entities involved can have some *preferences* for their possible partners. In general, a matching on a bipartite graph is said to be **stable** if no vertex is unmatched and there is no other possible partner outside of the matching that they would prefer to be matched with. The preferences of each vertex $v \in V(G)$ are described through a bijective weight function $w_v : V \to [|X|]$, where $X = B$ if $v \in A$ and $X = A$ otherwise.

Definition 2.5: Stable matching

Let G be a bipartite graph and let $\{w_v\}_{v \in V(G)}$ be a family of preference functions. A matching M on G is said to be **stable** if for all edges $ab \in E(G)$ at least one of the following holds:

- a is matched and $\nexists ab' \in M$ such that $w_a(b) > w_a(b')$
- b is matched and $\nexists a'b \in M$ such that $w_b(a) > w_a(a')$



$w_a : 4231$	$w_1 : acbd$
$w_b : 4123$	$w_2 : abdc$
$w_c : 1243$	$w_3 : bcda$
$w_d : 3241$	$w_4 : dcba$

Figure 2.12: A stable matching. The preferences are represented through ordered lists

We observe that the definition of stable matching implies that the trivial empty matching is not stable, since at least one of the two endpoints of every edge has to be matched. Surprisingly, Gale and Shapley [GS13] proved that in any bipartite graphs it is always possible to find a stable matching independently of all preferences and entities involved. The

result was first proved in the context of mathematical economics, solving the problem of optimal matching processes, such as college admission, job recruiting, hospital admitting and many more, awarding the two authors the *2012 Nobel Prize in Economics*. Before proving the result, we give some a "name" to some properties in order to make things more readable:

- Given two matchings M, M' on a graph G, we say that M' is **better** than M if for all $ab \in M'$ there is at least an edge $a'b \in M$ such that $w_b(a') \geq w_b(a)$ and there is at least an edge $a''b \in M'$ such that $w_b(a) > w_b(a'')$. In other words, there is at least one edge that is strictly more preferable, while all the other edges are at least as good as the ones in M.
- We say that a vertex $a \in A$ is **acceptable** for $b \in B$ if either b is unmatched or b is matched to $a' \in A$ but $w_b(a) > w_b(a')$. In other words, b is either unmatched or matched to a partner less preferred than a.
- We say that $a \in A$ is **happy** if either a is unmatched or a is matched to $b \in B$ and for all $b' \in B$ that find a acceptable it holds that $w_a(b) \geq w_a(b')$. In other words, a is either unmatched or has reached its optimal partner.

Theorem 2.5: Gale-Shapley theorem

Every bipartite graph with a family of preference functions $\{w_v\}_{v \in V(G)}$ has a stable matching.

Proof. We give a constructive proof. The idea is to construct a sequence of matchings $M_0, \ldots, M_k$ such that for each matching it holds that every vertex of A is happy inside it, but only the last matching is stable. First, we claim the following.

Claim 1: if M is not stable and every vertex of A is happy in M then there is no $a \in A$ such that a is unmatched and acceptable for some $b \in B$

Proof of Claim 1. Suppose that M is not stable and that every vertex of A is happy in M. Then, by definition of unstable matching, there must be at least one edge $ab \in E(G)$ for which one of the following four cases holds:

1. a is unmatched and b is unmatched. Then, a is acceptable for b since the latter is unmatched.
2. a is unmatched and $\exists a'b \in M$ such that $w_b(a) > w_b(a')$. Then, a is acceptable for b since the latter is matched to a' but a is a more preferred partner for b.
3. $\exists ab' \in M$ such that $w_a(b) > w_a(b')$ and b is unmatched. Then, a is acceptable to b since the latter is unmatched, contradicting the fact that a is happy in M since a is matched to b' but prefers b.
4. $\exists ab' \in M$ such that $w_a(b) > w_a(b')$ and $\exists a'b \in M$ such that $w_b(a) > w_b(a')$. Then, a is acceptable to b since a is a more preferred partner for b, contradicting the fact that a is happy in M since a is matched to b' but prefers b.

Hence, the only possibilities are the first and second case, concluding that a is unmatched and acceptable for b. □

We'll now define the extension procedure. Let $M_0 = \varnothing$. Since the matching is empty, each vertex of A is unmatched, hence happy. Moreover, by definition M_0 cannot be stable. This will act as our base case. Fix $i \in [k]$ and suppose that M_i is unstable and every vertex inside it is happy. Then, by Claim 1 there must be a vertex $a_{i+1} \in A$ that is unmatched in M_i and must have at least one $b \in B$ that finds a acceptable. Let b_{i+1} be a vertex of B that finds a acceptable and that maximizes the value of $w_{a+1}(b_{i+1})$. We set M_{i+1} as the matching obtained by removing the potential already existing partner a' of b_{i+1} in M_i and by adding the edge $a_{i+1}b_{i+1}$

$$M_{i+1} = \begin{cases} M_i \cup \{a_{i+1}b_{i+1}\} & \text{if } \nexists a'b_{i+1} \in M_i \\ (M_i \cup \{a_{i+1}b_{i+1}\}) - \{a'b_{i+1}\} & \text{otherwise} \end{cases}$$

We claim that the extended matching is better than the previous and preserves happiness for all vertices.

Claim 2: M_{i+1} is better than M_i and every vertex of A is happy in M_{i+1}.

Proof of Claim 2. By construction, every edge $xy \in M_i$ such that $y \neq b_{i+1}$ is also inside M_{i+1}, hence every such edge preserves happiness and the overall preferences of each vertex. If $\nexists a'b_{i+1} \in M_i$ then M_{i+1} is trivially better than M_i. If $\exists a'b_{i+1} \in M_i$, instead, the new edge $a_{i+1}b_{i+1}$ since a_{i+1} is acceptable for b, meaning that M_{i+1} is better than M_i also in this case. Moreover, by choice of b_{i+1}, the vertex a_{i+1} is now matched with an optimal partner. Hence, the happiness of a_{i+1} is preserved: a_{i+1} was happy due to it being unmatched, but now it's happy due to being matched with the best partner. Moreover, the happiness of the potential vertex a' is also happy since it became unmatched. □

Since every matching in the sequence is better than the previous ones, we know that such sequence cannot cycle. Moreover, since there is a finite number of possible matchings, the sequence will eventually reach a matching where every vertex of A is matched. Let M_k be such matching. Then, by contrapositive of Claim 1, we know that M_k is either stable or has a vertex that isn't happy. However, by construction of the sequence, we know that every vertex in M_k is happy, thus the only possibility is that M_k is stable. □

2.4 Solved exercises

Problem 2.1

Let G be a bipartite k-regular graph with bipartition (A, B). Prove that:

1. $|A| = |B|$
2. G has a perfect matching

Proof. Given any set X, let E_X denote the subset of edges with an endpoint in X. Since the graph is k-regular, we have that $|E_A| = k\,|A|$ and $|E_B| = k\,|B|$. Moreover, since the graph is bipartite, we have that $E_A = E_B$. Hence, we get that $k\,|A| = k\,|B|$ and thus that $|A| = |B|$. Consider now a subset $S \subseteq A$. We observe that $E_S \subseteq E_{N(S)}$ by definition of neighborhood. Hence, we have that $k\,|S| = |E_S| \leq \left|E_{N(S)}\right| = k\,|N(S)|$ and thus that $|S| \leq |N(S)|$. By Hall's marriage condition, we conclude that G has a perfect matching. □

3

Structure analysis

Disjoint paths, hitting set and separations

Many tools of graph theory enable us to study the structure of the underlying graph when some conditions are met. One such tool is **Menger's theorem**, which can be used to derive many results regarding the structure of graphs. We'll start by defining the tools needed to discuss such theorem.

Definition 3.1: A, B **path**

Let G be a graph. Given two subsets $A, B \subseteq V(G)$ is a path with one endpoint in A, one endpoint in B and no internal vertex in $A \cup B$

We're interested in finding the maximum number of disjoint A, B paths.

Bibliography

[Ber57] Claude Berge. "Two theorems in graph theory". In: *Proceedings of the National Academy of Sciences* (1957). DOI: 10.1073/pnas.43.9.842.

[Dir52] G. A. Dirac. "Some Theorems on Abstract Graphs". In: *Proceedings of the London Mathematical Society* (1952). DOI: 10.1112/plms/s3-2.1.69.

[Edm65] Jack Edmonds. "Paths, Trees, and Flowers". In: *Canadian Journal of Mathematics* 17 (1965). DOI: 10.4153/CJM-1965-045-4.

[Eul41] Leonhard Euler. "Solutio problematis ad geometriam situs pertinentis". In: *Commentarii academiae scientiarum Petropolitanae* (1741).

[GS13] D. Gale and L. S. Shapley. "College Admissions and the Stability of Marriage". In: *The American Mathematical Monthly* (2013). URL: https://www.jstor.org/stable/10.4169/amer.math.monthly.120.05.386.

[Hal35] P. Hall. "On Representatives of Subsets". In: *Journal of the London Mathematical Society* (1935). DOI: https://doi.org/10.1112/jlms/s1-10.37.26.

[HK73] John E. Hopcroft and Richard M. Karp. "An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs". In: *SIAM Journal on Computing* (1973). DOI: 10.1137/0202019.

[LP09] L. Lovász and M.D. Plummer. *Matching Theory*. AMS Chelsea Pub., 2009. URL: https://books.google.it/books?id=0aoJBAAAQBAJ.

[Sza20] Gabor Szarnyas. *Graphs and matrices: A translation of "Graphok és matrixok" by Dénes Kőnig (1931)*. 2020. URL: https://arxiv.org/abs/2009.03780.