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

Graph Theory

Lecture notes integrated with the book TODO

Author
Alessio Bandiera

March 7, 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/aflaag-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: **alessio.bandiera02@gmail.com**
- LinkedIn: **Alessio Bandiera**

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

Suggested prerequisites:

- Progettazione degli Algoritmi

Licence:

1

Basics of Graph Theory

placeholder

write a small introduction on graph theory

1.1 Introduction

Definition 1.1: Graph

A **graph** is a pair $G = (V, E)$, where V is the — finite — set of **vertices** of the graph, and E is the set of **edges**.

For now, will assume to be working with **simple** and **undirected** graphs, i.e. graphs in which the set of edges is defined as follows

$$E \subseteq [V]^2 = \{\{x, y\} \mid x, y \in V \wedge x \neq v\}$$

where the notation $\{x, y\}$ will be used to indicate an edge between two nodes $x, y \in V$, and will be replaced with $xy = yx$ directly — the *set* notation for edges is used to highlight that edges have no direction.

We will indicate with n and m the cardinality of $|V|$ and $|E|$, respectively. Moreover, we will indicate with $V(G)$ and $E(G)$ the set of the vertices and edges of G respectively when there is ambiguity.

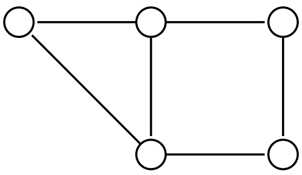


Figure 1.1: A simple graph.

Note that, in this definition, we are assuming that each edge has exactly 2 *distinct* endpoints — i.e. the graphs do not admit **loops** — and there cannot exist two edges with the same endpoints. In fact, if we drop these assumption we obtain what is called a **multigraph**.

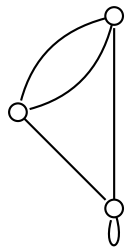


Figure 1.2: A multigraph.

Definition 1.2: Subgraph

Given a graph $G = (V, E)$, a **subgraph** $G' = (V', E')$ of G is a graph such that $V' \subseteq V$ and $E' \subseteq E$.

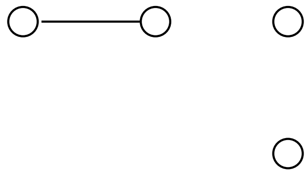


Figure 1.3: This is a subgraph of the graph shown in Figure 1.1.

Definition 1.3: Induced subgraph

Given a graph $G = (V, E)$, a subgraph $G' = (V', E')$ of G is **induced** if every edge of G with both ends in V is an edge of V'.

This definition is *stricter* than the previous one: in fact, the last graph is *not* an example of an induced subgraph, but the following is:

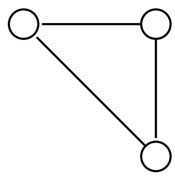
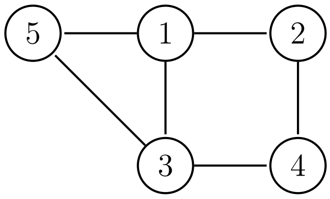


Figure 1.4: This is an *induced* subgraph of the graph shown in Figure 1.1.

Note that every induced subgraph of a graph is **unique** by definition, and we indicate each induced subgraph as follows: suppose that the graph in Figure 1.1 had the following *labeling* on the vertices



then, the induced subgraph in Figure 1.4 would have been referred to as $G[\{1,3,5\}]$.

Intuitively, two vertices $x, y \in V$ are said to be **adjacent**, if there is an edge $xy \in E$, and we write $x \sim y$. If there is no such edge, we write $x \nsim y$ for non-adjacency. The **neighborhood** of a vertex $x \in V$ is the set of vertices that are adjacent to x, and it will be indicated as follows

$$\mathcal{N}(x) := \{y \in V \mid x \sim y\}$$

The **degree** of a vertex $x \in V$, denoted with $\deg(x)$, is exactly $|\mathcal{N}(x)|$. We will use the following notation for the **minimum** and **maximum** degree of a graph, respectively

$$\delta := \min_{x \in V} \deg(x) \qquad \Delta := \max_{x \in V} \deg(x)$$

Lemma 1.1: Handshaking lemma

Given a graph $G = (V, E)$, it holds that

$$\sum_{x \in V} \deg(x) = 2\,|E|$$

Proof. Trivially, the sum of the degrees counts every edge in E exactly twice, once for each of the 2 endpoints. □

Definition 1.4: k-regular graph

A graph G is said to be **k-regular** if every vertex of G has degree k.

Note that in a k-regular graph it holds that

$$\sum_{x \in V} \deg(x) = k \cdot n$$

Proposition 1.1

There are no k-regular graphs with k odd and an odd number of vertices.

Proof. By way of contradiction, suppose that there exists a k-regular graph $G = (V, E)$ such that both k and n are odd; however, by the handshaking lemma we would get that

$$2\,|E| = \sum_{x \in V} \deg(x) = k \cdot n$$

but the product of two odd numbers, namely k and n, is still an odd number, while $2\,|E|$ must be even $\lightning$. $\square$

1.1.1 Important structures

Definition 1.5: Path

A **path** is a *graph* with vertex set $x_0, \dots, x_n$ and edge set $e_1, \dots, e_n$ such that $e_i = x_{i-1}x_i$.

The **length** of a path is the number of edges between x_0 and x_n, i.e. $|\{e_1, \dots, e_n\}|$, namely n in this case. A path of length 1 is called *trivial* path.

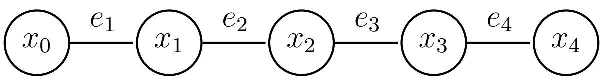


Figure 1.5: A path graph of length 4 that links x_0 and x_4.

Through *paths* we can provide the definition of **distance** between two nodes of a graph.

Definition 1.6: Distance

Given a graph $G = (V, E)$, and two vertices $x, y \in V$, the **distance** between x and y in G, denoted with $\mathrm{dist}_G(x, y)$, is defined as the length of the *shortest* path between x and y in G.

If there is no ambiguity, we will simply write $\mathrm{dist}(x, y)$ instead of $\mathrm{dist}_G(x, y)$. Finally, given a path P and two vertices $u, v \in V(P)$, we will denote with $u\ P\ v$ the *subpath* of P between u and v. Now, consider the following definition.

Definition 1.7: Walk

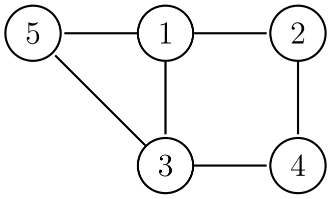
Given a graph $G = (V, E)$, a **walk** is a *sequence* of vertices and edges

$$x_0\ e_1\ x_1\ \dots\ x_{k-1}\ e_k\ x_k$$

where $x_0, \dots, x_k \in V$, $e_1, \dots, e_k \in E$ and $e_i = x_{i-1}x_i$.

The **length** of a walk is the number of edges between x_0 and x_k, i.e. $|\{e_1, \dots, e_k\}|$, namely k in this case. If $x_0 = x_k$ we say that the walk is **closed**.

If there is a path – or a walk — between two vertices $x, y \in V$, we say that the path — or the walk — **links** x and y, and we write this as $x \to y$. Any vertex that is different from x and y is called *internal node*. For instance, given the previous graph labeled as follows



an example of a walk over this graph is given by the following sequence

$$1 \ \{1,2\} \ 2 \ \{2,4\} \ 4 \ \{4,3\} \ 3 \ \{3,1\} \ 1 \ \{1,5\} \ 5$$

that *links* 1 and 5, i.e. the walk is of the form $1 \to 5$.

Note that there is a subtle difference between the definitions of **path** and **walk**: the definition of a path implies that this is always a *graph* on its own, while a walk is defined as a *sequence*. Nonetheless, we will treat *paths* as if they where *sequences* as well. This assumption holds for the following structures that will be discussed as well.

However, by definition of path, not every alternating sequence of vertices and edges is a valid path, in fact:

- in a *walk* it is possible to repeat both vertices and edges
- in a *path* there can be no repetition of vertices nor edges (note that *edge* repetition implies *vertex* repetition)

For instance, the previous example of *walk* is not a valid *path*, because the vertex 1 is repeated.

Theorem 1.1: Paths and walks

Given a graph $G = (V, E)$ and two vertices $x, y \in V$, in G there is a path $x \to y$ if and only if there is a walk $x \to y$.

Proof. By definition, every path is a walk, thus the direct implication is trivially true. To prove the converse implication, consider two vertices x and y for which there is at least one walk $x \to y$ in G. Now, out of all the possible walks $x \to y$ in G, consider the *shortest* one, i.e. the one with the least amount of edges, and let it be the following sequence

$$x \ e_1 \ x_1 \ \dots \ x_{k-1} \ e_k \ y$$

By way of contradiction, assume that this walk is not a path. Therefore, there must be either one vertex or one edge repeated, but since edge repetition always implies vertex repetition, we just need to take this case into account. Assume that there are two indices $i, j \in [k-1]$ such that $i \neq j$ and $x_i = x_j$; however this implies that

$$x \ e_1 \ \dots \ x_{i-1} \ e_i \ x_i \ e_{j+1} \ x_{j+1} \ \dots \ x_{k-1} \ e_k \ y$$

is still a walk $x \to y$ of strictly shorter length, but we chose the original sequence to be the *shortest* possible walk $x \to y$ $\lightning$. $\square$

Proposition 1.2

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

Proof. Consider a graph $G = (V, E)$, and let P be a longest path in G, labeled as follows

$$x_0 \ e_1 \ x_1 \ \dots \ x_{k-1} \ e_k \ x_k$$

and assume that its length is k. Since P is a longest path in G, x_k cannot have neighbors outside P itself, otherwise P would not have been the longest path of G — it could have been extended by one of x_k's neighbors. This implies that

$$\mathcal{N}(x_k) \subseteq \{x_0, \dots, x_{k-1}\}$$

and since $\delta \leq \deg(x_k) := |\mathcal{N}(x_k)|$ by definition of δ, this implies that

$$\delta \leq |\{x_0, \dots, x_{k-1}\}| = k$$

$\square$

Definition 1.8: Cycle

A **cycle** is a *graph* with vertex set $x_1, \dots, x_n$ and edge set $x_1x_2, x_2x_3, \dots, x_{n-1}x_n, x_nx_1$.

The **length** of a cycle is the number of edges between x_1 and x_n, namely n in this case. A cycle of length n is denoted as C_n.

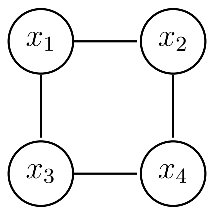


Figure 1.6: A cycle graph of length 4.

A graph that does not admit cycle subgraphs — or *cycles*, for short — is said to be **acyclic**.

Proposition 1.3

Every graph with $\delta \geq 2$ has a cycle of length at least $\delta + 1$.

Proof. Consider the proof of Proposition 1.2; by applying the same reasoning, we know that x_k cannot have neighbors outside P itself. However, since $\delta \geq 2$, and $x_k \sim x_{k-1}$, there must be at least one vertex in x_k's neighborhood that lies in P. Therefore, let x_i be the first vertex of P — w.r.t. our labeling of P — that is adjacent to x_k; hence, we have

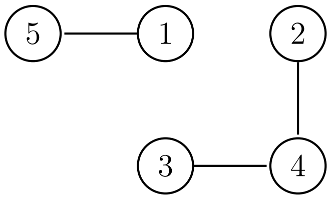
$$\mathcal{N}(x_k) \subseteq \{x_i, \dots, x_{k-1}\} \implies \delta \leq |\{x_i, \dots, x_{k-1}\}|$$

which implies that $x_i, \dots, x_{k-1}, x_k, x_i$ is a cycle of length at least $\delta + 1$. $\square$

Definition 1.9: Connected graph

An undirected graph $G = (V, E)$ is said to be **connected** if and only if for each vertex pair $x, y \in V$ there is a path $x \to y$.

All the graphs that we presented so far are *connected*, thus the following figure provides an example of an **disconnected** graph.



Definition 1.10: Component

Given a graph G, a **component** of G is a maximal connected subgraph of G.

For instance, the graph of the previous example is made up of 2 components, namely the following two subgraphs

$$C_1 = (\{1, 5\}, \{\{1, 5\}\})$$
$$C_2 = (\{2, 3, 4\}, \{\{2, 4\}, \{4, 3\}\})$$

Proposition 1.4

If G is a connected graph, and C is a cycle in G, then for any edge $e \in C$ it holds that $G - \{e\}$ is still connected.

Proof. Consider a graph $G = (V, E)$ that has a cycle C, and any two vertices $x, y \in V$; in particular, since G is connected, there must exist a path $x \to y$ in G, and let this path be

$$P = x \ e_1 \ x_1 \ \ldots \ x_{k-1} \ e_k \ y$$

Consider an edge $e \in C$; if P does not traverse e, trivially $G - \{e\}$ will still contain P.

Now let

$$C = z_1 \ f_2 \ z_2 \ \ldots \ z_{l-1} \ f_l \ z_l \ f_{l+1} \ z_1$$

and w.l.o.g. assume that $e = f_2 = z_1 z_2 = x_i x_{i+1} = e_{i+1}$ for some $i \in [k-1]$. Thus we can construct the following walk

$$x \ e_1 \ x_1 \ \ldots \ x_i \ f_{l+1} \ z_l \ \ldots \ f_3 \ z_2 \ e_{i+2} \ x_{i+2} \ \ldots \ x_{k-1} \ e_k \ y$$

from x to y, and by Theorem 1.1 we have that there is a path from $x \to y$, which proves that $G - \{e\}$ is still connected. $\square$

Definition 1.11: Tree

A **tree** is a connected acyclic graph. Usually, but not necessarily, there is a fixed vertex called **root**, and any vertex that has degree 1 in the tree is called **leaf**.

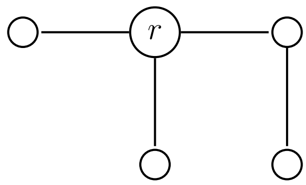


Figure 1.7: A tree with tree leaves, rooted in r.

A **forest** is an disconnected graph in which each component is a *tree*, as in the following example

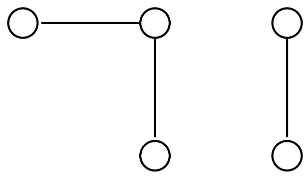


Figure 1.8: A forest.

Given a tree T rooted in some node $r \in V(T)$, and two vertices $x, y \in V(T)$, consider the paths P_x of the form $x \to r$ and P_y of the form $y \to r$, respectively. The first vertex of P_y that is encountered by tracing P_x from x to r is called **lowest common ancestor** (LCA) of x and y.

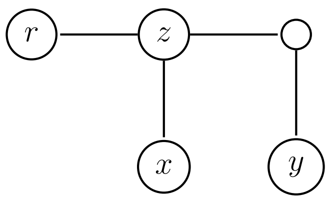


Figure 1.9: For instance, in this tree — rooted in r — the LCA of x and y is the vertex z.

Note that the LCA between any two vertices of a tree is *always defined*, since in the "worst case" it is the root r itself.

Theorem 1.2: Alternative definitions of tree

Given a graph $T = (V, E)$, the following statements are equivalent:

1. T is a tree
2. every vertex pair of T is connected by a unique path
3. T is *minimally connected*, i.e. T is connected and $\forall e \in E$ it holds that $T - \{e\}$ is disconnected
4. T is *maximally acyclic*, i.e. T is acyclic and $\forall x, y \in V$ such that $x \nsim y$, it holds that $T \cup \{xy\}$ has a cycle

Proof. We will prove the statements cyclically.

- $1 \implies 2$. By contrapositive, assume that in T there exist two vertices $x, y \in V$ for which there are two distinct paths P and Q of the form $x \to y$. If P and Q are edge-disjoint, then $P \cup Q$ is a cycle, which implies that T is not a tree by definition.

  Otherwise, assume that P and Q are not edge-disjoint. If we start say in x, and we follow Q edge by edge since P and Q are distinct, at some point we will encounter an edge $\{u, v\}$ such that $u \in P \cap Q$ and $v \in Q - P$ — possibly, $u = x$ itself. Moreover, since both paths lead to y, if we keep following Q we will encounter a vertex $z \in P \cap Q$ — possibly, $z = y$ itself — from which the two paths will coincide. Let Q' be the subpath of P starting with u and ending in z; then, $Q' \cup (Q - P)$ is a cycle in T, which implies that T is not a tree by definition.

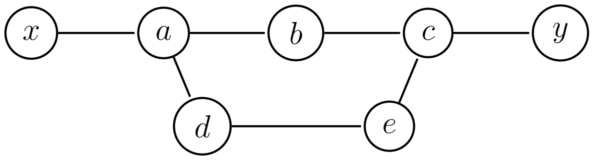


Figure 1.10: For instance, applying the argument of the proof in this graph we would get that $P = \{x, a, b, c, y\}$, $Q = \{x, a, d, e, c, y\}$, $Q - P = \{d, e\}$, $u = a$, $z = c$ and $Q' = \{a, b, c\}$, in fact $Q' \cup (Q - P) = \{a, b, c, e, d\}$ which is a cycle.

- $2 \implies 3$. Consider an edge $xy \in E$; this edge itself is a path $x \to y$, and if we assume statement 2 this implies that it is the *only* path from x to y. This implies that $T - \{xy\}$ cannot contain a path from x to y, therefore $T - \{xy\}$ is disconnected.
- $3 \implies 4$. Since statement 3 implies that T is connected, by Proposition 1.4 we have that T is acyclic. Now, pick $x, y \in V$ such that $x \nsim y$; by connectivity of T there must be a path $x \to y$ in T, and let this path be P. Lastly, since $x \nsim y$, we have that $P \cup \{xy\}$ is a cycle in T.
- $4 \implies 1$ By contrapositive, we want to prove that if T is not a tree, then T is not maximally acyclic. Note that if T is not a tree, we have two options:
  - if T is connected but contains a cycle, then T is clearly not maximally acyclic

- if T is acyclic but disconnected, then by definition there must exist two vertices $x, y \in V$ such that there is no path $x \to y$, which implies that $T \cup \{xy\}$ still does not contain any cycle

$\square$

Lemma 1.2

Every tree with at least 2 vertices has a leaf.

Proof. By way of contradiction, assume T is a tree with at least 2 vertices that does not contain any leaves; then $\delta \geq 2$ in T, which implies that T contains a cycle of length at least $\delta + 1$ by Proposition 1.3. $\square$

Lemma 1.3

Given a tree T, and a leaf v of T, it holds that $T - \{v\}$ is still a tree.

Proof. Since T is acyclic by definition, $T - \{v\}$ is still acyclic, we just need to prove that $T - \{v\}$ is still connected. By way of contradiction, assume that in $T - \{v\}$ there exist two vertices x and y such that there is no path between them. However, since T is connected, there is a path P of the form $x \to y$ in T.

Note that, if by removing v from T we disconnect x and y, it must be that v lies in P. Moreover, since v is in T but not in $T - \{v\}$, while both x and y are also in $T - \{v\}$, it must be that v is an *internal* node of P, i.e. $v \neq x, y$, which implies that $\deg(v) \geq 2$ by definition of path, contradicting the hypothesis for which v was a leaf $\lightning$. $\square$

Proposition 1.5

If T is a tree, then $m = n - 1$.

Proof. We will prove the statement by induction on n

Base case. When $n = 1$, there are no edges in the tree, and $0 = m = 1 - 1$.

Inductive hypothesis. Assume that for a tree that has $n = k - 1$ nodes the statement holds.

Inductive step. We will prove the statement for a tree T that has $n = k$ nodes. Note that, since $n = 1$ is the base case, we can assume that $n = k \geq 2$, hence by Lemma 1.2 T contains at least one leaf. Let this leaf be v; then, by Lemma 1.3 it holds that $T - \{v\}$ is still a tree, and clearly $T - \{v\}$ has $k - 1$ nodes, which implies that we can apply the inductive hypothesis on $T - \{v\}$, i.e.

$$|E(T - \{v\})| = |V(T - \{v\})| - 1 = k - 1 - 1 = k - 2$$

However, note that v is a leaf, concluding that

$$m = |E(T)| = |E(T - \{v\})| + 1 = k - 2 + 1 = k - 1 = n - 1$$

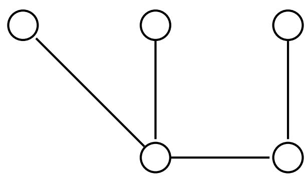
□

Definition 1.12: Spanning tree

Given a graph $G = (V, E)$, a **spanning tree** T of G is a subgraph of G such that

- T is a tree
- $V(T) = V(G)$, i.e. T *spans* every vertex of G

For instance, given the graph in Figure 1.1, a possible spanning tree is the following:



Lemma 1.4

Any connected graph has a spanning tree.

Proof. Consider a connected graph G, and keep removing edges from $E(G)$ — and their relative endpoints from $V(G)$ — one by one, as long as G is still connected. If no other edge can be removed from G without violating connectivity, we will end up with a graph that must be a tree by statement 3 of Theorem 1.2. □

Thanks to this last proposition, we can actually prove a stronger version of the Proposition 1.5, which is the following.

Theorem 1.3

T is a tree if and only if T is connected and $m = n - 1$.

Proof. The direct implication is proved in Proposition 1.5, so we just need to prove the converse implication. Consider a connected graph T such that $m = n - 1$; by Lemma 1.4 T must have a spanning tree T', and by Proposition 1.5 itself it holds that $|E(T')| = |V(T')| - 1$. However, T' is a spanning tree of T, therefore

$$V(T) = V(T') \implies |V(T)| = |V(T')| \implies |E(T')| = |V(T)| - 1 = |E(T)|$$

which implies $E(T) = E(T')$ because T' is a subgraph of T, therefore $T = T'$, concluding that T must be a tree since T' is a tree. □

1.1.2 Bipartite graphs

Definition 1.13: Bipartite graph

A graph $G = (V, E)$ is said to be **bipartite** if there exists a set $X \subseteq V$ such that every edge of G has exactly one endpoint in X and one in $V - X$. If such a set X exists, we say that $(X, V - x)$ is a **bipartition** of G.

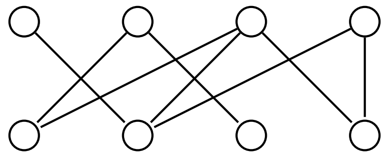


Figure 1.11: An example of a *bipartite graph*. In particular, if we call the "left" set of nodes A and the "right" one B, then (A, B) is a bipartition of the graph.

There are various types of graphs that can be bipartitioned. For example, every **tree** T can be bipartitioned through a bipartition $(X, V(T) - X)$ by considering the following set of vertices

$$X := \{v \in V(T) \mid \text{dist}(r, v) \text{ even}\}$$

where r is T's root.

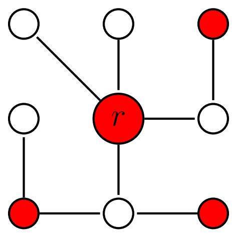


Figure 1.12: The set of red vertices X defines a bipartition $(X, V(T) - X)$ of this tree T.

However, *not every type* of graph can be bipartitioned. For instance, consider the following type.

Definition 1.14: Clique

A **clique** is a *graph* in which each vertex is adjacent to any other vertex of the graph. A clique that has n vertices is denoted as K_n.

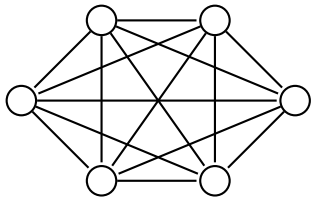


Figure 1.13: The clique K_6.

It is easy to see that no clique K_n can be bipartitioned, since there is an edge between any pair of vertices of the graph. However, this is not the only type of graph that cannot be bipartitioned.

Lemma 1.5

If G is a bipartite graph, and H is a subgraph of G, then H must be bipartite.

Proof. Given a bipartite graph $G = (V, E)$, assume that $(X, V - X)$ is a bipartition of G, and let H be a subgraph of G; then, it is easy to see that $(X \cap V(H), V(H) - X)$ is a bipartition for H. □

Note that this lemma implies that G is bipartite if and only if every connected component of G is bipartite: in fact, the direct implication follows from this lemma, and the following figure provides an intuition for the converse implication.

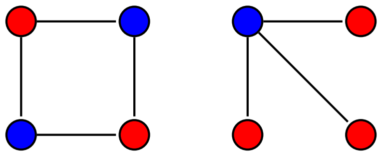


Figure 1.14: Consider the following disconnected graph G made of these two connected components, C_1 and C_2 respectively. Say that C_1 has a bipartition $(X_1, V(C_1) - X_1)$ where X_1 is the red set, and C_2 has a bipartition $(X_2, V(C_2) - X_2)$ where X_2 is the red set; thus, $(X_1 \cup X_2, V(C_1 \cup C_2) - X_1 - X_2)$ is clearly a bipartition of G. This process may be repeated for all the connected components of any disconnected graph.

Theorem 1.4: Bipartite graphs

G is bipartite if and only if G has no odd-length cycle.

Proof.

Direct implication. We will prove the contrapositive, i.e. if G has an odd-length cycle, then G cannot be bipartitioned. Consider a graph G with an odd-length cycle C_{2k+1} of vertices $x_1, \ldots, x_{2k+1}$; by way of contradiction, assume that G is bipartite through a bipartition $(X, V(G)) - X)$ for some $X \subseteq V(G)$. W.l.o.g. assume that $x_1 \in X$; then, since X defines a bipartition of G it must be that $x_2 \notin X$, and

$x_3 \in X$ and so on and so forth. In particular, for any odd value of i we will have that $x_i \in X$, but this implies that both x_1 and x_{2k+1} must be inside X, which means that the edge $x_1 x_{2k+1}$ violates the bipartition induced by X $\lightning$.

Converse implication. Again, we will prove the contrapositive, i.e. if G is not bipartite it must contain an odd-length cycle. By the previous observation, G is not bipartite if and only if at least one connected component of G is not bipartite, and let this component be $\overline{G}$. Note that, since $\overline{G}$ is connected, it must contain a spannning tree T by Lemma 1.4. Moreover, as previously described, we can always define a bipartition on a tree, namely $(X, V(T) - X)$ where

$$X := \{v \in V(T) \mid \mathrm{dist}_T(r, v) \text{ even}\}$$

for some root node $r \in V(T)$.

Now, since T is a spanning tree of $\overline{G}$, which is now bipartite by hypothesis, there must exist an edge $xy \in V(\overline{G})$ such that either $x, y \in X$ or $x, y \in V(T) - X$, i.e. the edge xy must have both endpoints in the same set of T's bipartition. Let z be the LCA between x and y in T, and P_x and P_y be the paths of the form $x \to r$ and $y \to r$, respectively. Note that, since xy has both endpoints in the same set, it must be that the lengths of P_x and P_y have the same parity by definition of X. Lastly, by statement 2 of Theorem 1.2 we have that $r\ P_x\ z = r\ P_y\ z$, which implies that the lengths of $z\ P_x\ x$ and $z\ P_y\ y$ must have the same parity. This concludes that

$$z\ P_x\ x \cup z\ P_y\ y \cup xy$$

is an odd-length cycle of G.

$\square$

1.2 Exercises

Problem 1.1

Let $G = (V, E)$ be a graph of n vertices, where $n \geq 2$. Show that there must exists two vertices $x, y \in V$ such that $\deg(x) = \deg(y)$.

Solution. By definition, the range of the possible degrees for any node of G is $[0, n-1]$. By way of contradiction, assume that for any two vertices $x, y \in V$ it holds that $\deg(x) \neq \deg(y)$; hence, since the graph has n nodes, it must be that each node is assigned a different degree, and that we use all the possible degrees in $[0, n-1]$. In particular, this implies that there are two vertices $u, v \in V$ such that $\deg(u) = 0$ and $\deg(v) = n - 1$, but this is a contradiction because if the degree of v is $n - 1$, it must be adjacent to all the other nodes of V, including u, and $\deg(u) = 0$ $\lightning$.