Chapter 13

Asymptotic Notation

13.1 Definitions

Analysis of algorithms is concerned with estimating how many steps various algorithms make while solving problems of various sizes. In particular, given an algorithm, we want to make statements like "For input of size n, the algorithm will terminate in at most $f(n)$ steps." If we try to accurately estimate the number of steps, a cumbersome bound like

$$f(n) = \frac{1}{11}n^3 + 12n^2 + 15\frac{1}{2}n + \log_3 n + 17$$

might arise. Such precision only complicates matters and does not add to our understanding of the algorithm's efficiency. The following notational convention allows to simplify bounds by concentrating on their "main terms."

Definition 13.1.1. *For two functions $f, g : \mathbb{N}^+ \to \mathbb{R}$,*

- *$f(n) = O(g(n))$ if and only if there exists a positive constant $c \in \mathbb{R}$ and a constant $n_0 \in \mathbb{N}$, such that $|f(n)| \leq c|g(n)|$ for all $n \geq n_0$.*
- *$f(n) = \Omega(g(n))$ if and only if $g(n) = O(f(n))$.*
- *$f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.*

Asymptotic notation does wonders to the above ugly bound: We can now say that $f(n) = \Theta(n^3)$, which makes it easier to see how the number of steps performed by the algorithm grows as n gets larger and larger. Notice how the asymptotic notation swallowed all the constants and lower-order terms! To prove that $f(n) = \Theta(n^3)$ we need to show that there exist positive constants $c_1, c_2 \in \mathbb{R}$ and a constant $n_0 \in \mathbb{N}$, such that $c_1 n^3 \leq f(n) \leq c_2 n^3$ for all $n \geq n_0$. (We dropped the absolute values that come from Definition 13.1.1 since $f(n)$ and n^3 are nonnegative for $n \in \mathbb{N}^+$.) We can take $n_0 = 1$, $c_1 = \frac{1}{11}$, and $c_2 = 45.6$. For the lower bound, clearly $f(n) \geq \frac{1}{11}n^3$ when $n \in \mathbb{N}^+$. For the upper bound, note that in this range $n^3 \geq n^2 \geq n \geq \log_3 n$, and $n^3 \geq 1$. All these inequalities can be proved by elementary algebraic manipulation. Thus we get

$$f(n) \leq \frac{1}{11}n^3 + 12n^3 + 15\frac{1}{2}n^3 + n^3 + 17n^3 \leq 45.6n^3.$$

We can also perfectly well say that $f(n) = O(n^4)$ or that $f(n) = O(n^{25})$; these bounds are considerably less informative but correct. On the other hand, the bound $f(n) = O(n^2)$ (or even $f(n) = O(n^{2.99})$) is *not* correct. Indeed, we have seen that $f(n) \geq \frac{1}{11}n^3$. On the other hand, for any positive constant $c \in \mathbb{R}$, $\frac{1}{11}n^3 \geq cn^2$ for all $n \geq 11c$. Thus there is no positive constant $c \in \mathbb{R}$ and a constant $n_0 \in \mathbb{N}$ so that $f(n) \leq cn^2$ for all $n \geq n_0$.

Asymptotic notation is asymmetric, so we never write a statement like $O(g(n)) = f(n)$; the O, Ω, and Θ are always present on the right side of the equality sign. (However, we can write $n^2 + O(n) = \Theta(n^2)$, for example.) The right way to think of statements like $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ is as inequalities; always remember what the notation means according to Definition 13.1.1.

13.2 Examples and properties

The following asymptotic inequalities can all be easily proved and are very useful. Do the proofs as an exercise. You might find induction or tools from elementary calculus helpful for some of these. You'll also need simple properties of logarithms, like the identity

$$\log_a n = \frac{\log_b n}{\log_b a}.$$

- For two constants $u, v \in \mathbb{R}$, if $u < v$ then $n^u = O(n^v)$. ("A bigger power swallows a smaller one.")
- If $f(n)$ is a degree-d polynomial in n then $f(n) = O(n^d)$. If the coefficient of n^d in $f(n)$ is nonzero then $f(n) = \Theta(n^d)$.
- For any real constants $b > 1$ and p, $n^p = O(b^n)$. ("An exponential swallows a power.")
- For any real constants $q > 0$ and p, $(\ln n)^p = O(n^q)$. ("A power swallows a logarithm.")
- For any real constants $a, b > 1$, $\log_a n = \Theta(\log_b n)$. This implies that we can write bounds like $O(\log n)$, $O(n \log n)$, etc., without specifying the base of the logarithm. ("Asymptotic notation swallows bases of logarithms.")

We conclude this lecture by demonstrating how new asymptotic inequalities can be derived from existing ones. These are often used in the analysis of algorithms, although they are so much a part of the folklore that they are rarely referred to explicitly.

Proposition 13.2.1. *The following hold:*

(a) If $f(n) = O(g(n))$ and $p \in \mathbb{N}$ is a constant then $p \cdot f(n) = O(g(n))$.

(b) If $f(n) = O(h(n))$ and $g(n) = O(w(n))$ then $f(n)+g(n) = O(\max(|h(n)|, |w(n)|))$.

(c) If $f(n) = O(h(n))$ and $g(n) = O(w(n))$ then $f(n) \cdot g(n) = O(h(n) \cdot w(n))$.

Proof. We prove each claim individually.

(a) If $f(n) = O(g(n))$ then there exists a positive constant $c \in \mathbb{R}$ and a constant $n_0 \in \mathbb{N}$, such that $|f(n)| \leq c|g(n)|$ for all $n \geq n_0$. Thus for $p \in \mathbb{N}$, $|p \cdot f(n)| = p|f(n)| \leq (pc)|g(n)|$ for all $n \geq n_0$, and by Definition 13.1.1, $p \cdot f(n) = O(g(n))$.

(b) If $f(n) = O(h(n))$ and $g(n) = O(w(n))$ then there exist two positive constants $c_1, c_2 \in \mathbb{R}$ and constants $n_1, n_2 \in \mathbb{N}$, such that $|f(n)| \leq c_1|h(n)|$ for all $n \geq n_1$ and $|g(n)| \leq c_2|w(n)|$ for all $n \geq n_2$. Then

$$|f(n)+g(n)| \leq |f(n)|+|g(n)| \leq c_1|h(n)|+c_2|w(n)| = (c_1+c_2)\max(|h(n)|, |w(n)|)$$

for all $n \geq \max(n_1, n_2)$, and by Definition 13.1.1, $f(n)+g(n) = O(\max(|h(n)|, |w(n)|))$.

(c) If $f(n) = O(h(n))$ and $g(n) = O(w(n))$ then there exist two positive constants $c_1, c_2 \in \mathbb{R}$ and constants $n_1, n_2 \in \mathbb{N}$, such that $|f(n)| \leq c_1|h(n)|$ for all $n \geq n_1$ and $|g(n)| \leq c_2|w(n)|$ for all $n \geq n_2$. Then

$$|f(n) \cdot g(n)| = |f(n)| \cdot |g(n)| \leq (c_1|h(n)|) \cdot (c_2|w(n)|) = (c_1c_2)|h(n) \cdot w(n)|$$

for all $n \geq \max(n_1, n_2)$, and by Definition 13.1.1, $f(n) \cdot g(n) = O(h(n) \cdot w(n))$.

□

Chapter 14

Graphs

14.1 Introduction

A *graph* G is an ordered pair (V, E), where V is a set and E is a set of two-element subsets of V. That is,

$$E \subseteq \big\{\{x, y\} : x, y \in V, x \neq y\big\}.$$

Elements of V are the *vertices* (sometimes called *nodes*) of the graph and elements of E are the *edges*. If $e = \{x, y\} \in E$ we say that x and y are *adjacent* in the graph G, that y is a *neighbor* of x in G and vice versa, and that the edge e is *incident* to x and y.

What are graphs good for? Graphs are perhaps the most pervasive abstraction in computer science. It is hard to appreciate their tremendous usefulness at first, because the concept itself is so elementary. This appreciation comes through uncovering the deep and fascinating theory of graphs and its applications.

Graphs are used to model and study transportation networks, such as the network of highways, the London Underground, the worldwide airline network, or the European railway network; the 'connectivity' properties of such networks are of great interest. Graphs can also be used to model the World Wide Web, with edges corresponding to hyperlinks; Google uses sophisticated ideas from graph theory to assign a PageRank to every vertex of this graph as a function of the graph's global properties. In this course we will introduce the basic concepts and results in graph theory, which will allow you to study and understand more advanced techniques and applications in the future.

14.2 Common graphs

A number of families of graphs are so common that they have special names that are worth remembering:

Cliques. A graph on n vertices where every pair of vertices is connected is called a *clique* (or n-clique) and is denoted by K_n. Formally, $K_n = (V, E)$, where $V = \{1, 2, \ldots, n\}$ and $E = \{\{i, j\} : 1 \leq i < j \leq n\}$. The number of edges in K_n is $\binom{n}{2}$.

Paths. A *path* on n vertices, denoted by P_n, is the graph $P_n = (V, E)$, where $V = \{1, 2, \ldots, n\}$ and $E = \{\{i, i+1\} : 1 \leq i \leq n-1\}$. The number of edges in P_n is $n-1$. The vertices 1 and n are called the *endpoints* of P_n.

Cycles. A *cycle* on $n \geq 3$ vertices is the graph $C_n = (V, E)$, where $V = \{1, 2, \ldots, n\}$ and $E = \{\{i, i+1\} : 1 \leq i \leq n-1\} \cup \{\{1, n\}\}$. The number of edges in C_n is n.

14.3 Some important concepts

Graph isomorphism. If the above definition of a cycle is followed to the letter, a graph is a cycle only if its vertices are natural numbers. So, for example, the graph $G = (V, E)$ with $V = \{A, B, C\}$ and $E = \{\{A, B\}, \{B, C\}, \{C, A\}\}$ would not be a cycle. This seems wrong, because G "looks like" a cycle, and for all practical purposes it is exactly like C_3. The concept of *graph isomorphism* provides a way to formally say that C_3 and G are "the same."

Definition 14.3.1. *Two graphs $G = (V, E)$ and $G' = (V', E')$ are said to be* isomorphic *if there exists a bijection $f : V \to V'$ such that*

$$\{x, y\} \in E \text{ if and only if } \{f(x), f(y)\} \in E'.$$

In this case we write $G \equiv G'$ and the function f is called an isomorphism *of G and G'.*

We generally regard isomorphic graphs to be essentially the same, and sometimes do not even draw the distinction. Hence graphs that are isomorphic to cliques, cycles and paths are themselves said to be cliques, cycles and paths, respectively.

Size. The number of edges of a graph is called it *size*. The size of an n-vertex graph is at most $\binom{n}{2}$, achieved by the n-clique.

Degree. The *degree* (or *valency*) of a vertex v in a graph $G = (V, E)$, denoted by $d_G(v)$, is the number of neighbors of v in G. More formally, this degree is

$$d_G(v) = \big|\{u \in V : \{v, u\} \in E\}\big|.$$

A graph in which every vertex has degree k is called *k-regular* and a graph is said to be *regular* if it is k-regular for some k.

The following is sometimes called the Handshake lemma. It can be interpreted as saying that the number of people at a cocktail party who shake hands with an odd number of others is even.

Proposition 14.3.2. *The number of odd-degree vertices in a graph is even.*

Proof. For a graph $G = (V, E)$, consider the sum of the degrees of its vertices:

$$s = \sum_{v \in V} d_G(v).$$

Observe that this sum counts every edge e twice, once for each of the vertices incident to e. Thus $s = 2|E|$, and, in particular, s is even. Subtracting from s the degrees of even-degree vertices of G, we see that the resulting quantity is the sum of the degrees of odd-degree vertices and is still even. This implies the proposition. □

Subgraphs and Connectivity.

Definition 14.3.3. *Given a graph $G = (V, E)$,*

- *A graph $G' = (V', E')$ is said to be a* subgraph *of G if and only if $V' \subseteq V$ and $E' \subseteq E$.*
- *A graph $G' = (V', E')$ is said to be an* induced subgraph *of G if and only if $V' \subseteq V$ and $E' = \{\{u, v\} \in E : u, v \in V'\}$.*

Given a graph G, a path, cycle, or clique in G is a subgraph of G that is a path, cycle, or clique, respectively. Two vertices v and u of G are said to be *connected* if and only if there is a path in G with endpoints u and v. A graph G as a whole is said to be connected if and only if every pair of vertices in G is connected.

A subgraph G' of G is called a *connected component* of G if it is connected and no other graph G'', such that $G' \subset G'' \subseteq G$, is connected. Clearly, a graph is connected if and only if it has a single connected component.

Finally, there is a related notion to a path that is also useful: Given a graph $G = (V, E)$, a *walk* W in G is a sequence $W = (v_1, e_1, v_2, e_2, \ldots, v_{n-1}, e_{n-1}, v_n)$ of vertices and edges in G that are not necessarily distinct, such that $\{v_1, v_2, \ldots, v_n\} \subseteq V$, $\{e_1, e_2, \ldots, e_{n-1}\} \subseteq E$, and $e_i = \{v_i, v_{i+1}\}$ for all $1 \leq i \leq n-1$. A walk differs from a path in that vertices and edges can be repeated. The set of edges $\{e_1, e_2, \ldots, e_{n-1}\}$ covered by W is denoted by $E(W)$. Similarly, the set of vertices covered by W is $V(W) = \{v_1, v_2, \ldots, v_n\}$.

14.4 Kinds of graphs

What we have been calling graph is actually only one of many kinds of graphs, namely an undirected, unweighted, simple graph. Let's see how each of these qualities can differ and what other kinds of graphs there are.

A *directed* (simple, unweighted) graph G is an ordered pair (V, E), where V is a set and E is a set of ordered pairs from V. That is,

$$E \subseteq \big\{(x, y) : x, y \in V, x \neq y\big\}.$$

Directed graphs are suitable for modeling one-way streets, non-reflexive relations, hyperlinks in the World Wide Web, and so on. The notion of degree as defined above

is no longer applicable to a directed graph. Instead, we speak of the *indegree* and the *outdegree* of a vertex v in G, defined as $|\{u \in V : (u, v) \in E\}|$ and $|\{u \in V : (v, u) \in E\}|$, respectively.

A graph that is not simple can have *multi-edges* and *self-loops*. Multi-edges are multiple edges between the same pair of vertices. (Their presence means that the collection of edges is no longer a set, but a so-called *multiset*.) A self-loop is an edge to and from a single vertex v.

Finally, a graph can also be *weighted*, in the sense that numerical weights are associated with edges. Such weights are extremely useful for modeling distances in transportation networks, congestion in computer networks, etc. We will not dwell on weighted graphs in this course. In fact, unless specified otherwise, the word "graph" will continue to refer to undirected, unweighted, simple graphs.

Chapter 15

More Graphs—Eulerian, Bipartite, and Colored

15.1 Eulerian graphs

Ever seen those puzzles that ask you to trace some shape without lifting the pencil off the paper? For graph theory initiates such questions present no difficulty, separating this select elite from the rest of the human race who are doomed to spend their Sunday afternoons hunched over, putting page after page out of commission, searching in vain for the ever-elusive drawing.

Given a graph $G = (V, E)$, define a *tour* of G as a walk $T = (v_1, e_1, v_2, e_2, \ldots, v_n, e_n, v_{n+1})$ in G, such that T does not trace any edge more than once. (That is, $e_i \neq e_j$ for all $1 \leq i < j \leq n$.) The tour is said to be *Eulerian* if, in addition, $v_{n+1} = v_1$, $V(T) = V$, and $E(T) = E$. Thus an Eulerian tour traverses all the edges of G, "walking along" each exactly once, eventually coming back to where it started. (Particular vertices may and generally will be visited more than once.) A graph is said to be Eulerian if and only if it has an Eulerian tour.

Eulerian graphs were discussed by the great Leonhard Euler, the most prolific mathematician of all time. Euler's analysis of these graphs, presented in 1736, marks the birth of graph theory.

Theorem 15.1.1. *A graph is Eulerian if and only if it is connected and each of its vertices has even degree.*

Proof. We first prove that if G is Eulerian its vertices all have even degree. Indeed, trace an Eulerian tour of G starting and ending at a vertex v. Every time the tour enters an intermediate vertex it also leaves it along a different edge. In the very first step the tour leaves v and in the last step it enters v. Thus we can label the edges incident to any vertex as "entering" and "leaving", such that there is a bijection between these two sets. This shows that the degree of every vertex is even.

To prove that a graph $G = (V, E)$ with all vertex degrees being even is Eulerian, consider the longest tour $T = (v_1, e_1, v_2, e_2, \ldots, v_n, e_n, v_{n+1})$ in G. (The length of a tour is measured by its number of edges.) We prove below that T is Eulerian. Namely, we prove that:

(a) $v_1 = v_{n+1}$

(b) $n = |E|$

Proof of (a). Assume for the sake of contradiction that $v_1 \neq v_{n+1}$. Then the number of edges of T incident to v_1 is odd. (After T first leaves v_1, it enters and leaves it an even number of times.) Since the degree of v_1 in G is even, there is an edge e of G that is incident to v_1 but not part of T. We can extend T by this edge, obtaining a contradiction.

Proof of (b). We can assume that $v_1 = v_{n+1}$. Suppose $V(T) \neq V$. Consider a vertex $v \in V \setminus V(T)$ and a vertex $u \in V(T)$. Since G is connected, there is a path P between v and u in G. Consider the first time a vertex of T is encountered along P; this vertex is v_i for some $1 \leq i \leq n$. Let $e' = \{v', v_i\}$ be the edge along which P arrives at v_i and note that $v' \notin V(T)$. This implies that we can augment T by v' and e', and obtain a longer tour T', namely

$$T' = (v', e', v_i, e_i, \ldots, v_n, e_n, v_1, e_1, \ldots, v_{i-1}, e_{i-1}, v_i).$$

We have reached a contradiction and can therefore assume that $V(T) = V$. That is, T visits all the vertices of G. Assume for the sake of contradiction that $E(T) \neq E$, so there exists an edge $e' = \{v_i, v_j\}$ of G, for some $1 \leq i < j \leq n$, that is not part of T. Then we can augment T by the edge e', and obtain a longer tour T', namely

$$T' = (v_i, e', v_j, e_j, v_{j+1}, e_{j+1}, \ldots, v_n, e_n, v_1, e_1, \ldots, v_i, e_i, \ldots, v_{j-1}, e_{j-1}, v_j).$$

T' is longer than T by one edge, which is a contradiction that proves the theorem.

□

Proof technique: Considering an extremal configuration. In the above proof the crucial idea was to consider the longest tour in the graph. This is an instance of a common proof technique: If we need to prove that some configuration with particular properties exists (like an Eulerian tour), consider the *extremal* (longest, shortest, etc.) configuration of a related type (usually one that has some but not all of the required properties), and prove that this extremal configuration has to satisfy *all* of the required properties. Some steps in the proof usually proceed by contradiction: If the extremal configuration wasn't of the required type we could find a "more extremal" one, which is a contradiction.

15.2 Graph coloring

Consider a wireless company that needs to allocate a transmitter wavelength to each of its users. Two users who are sufficiently close need to be assigned different wavelengths to prevent interference. How many different wavelengths do we need? Of

course, we can just assign a new wavelength to every user, but that would be wasteful if some users are far apart. So what's the least number of wavelengths we can get away with?

We can model the users as vertices in a graph and connect two vertices by an edge if the corresponding users are sufficiently close. A *coloring* of this graph $G = (V, E)$ is an assignment of colors to vertices, such that no two adjacent vertices get the same color. The above question can now be restated as asking for the minimum number of colors that are needed for a coloring of G.

Let us be a bit more precise in defining colorings: A *k-coloring* of G is said to be a function $c : V \to \{1, 2, \ldots, k\}$, such that if $\{v, u\} \in E$ then $c(v) \neq c(u)$. The smallest $k \in \mathbb{N}$ for which a k-coloring of G exists is called the *chromatic number* of G. If a k-coloring of G exists, the graph is said to be k-colorable. There are many deep results concerning colorings and the chromatic number. At this point we only give the simplest one:

Proposition 15.2.1. *If the degree of every vertex in a graph G is at most k, then the chromatic number of G is at most $k + 1$.*

Proof. By induction on the number of vertices in G. (The degree bound k is fixed throughout the proof.) If G has a single vertex, then the maximal degree is 0 and the graph is 1-colorable. Since $1 \leq k+1$, the proposition holds. Suppose every graph with at most n vertices and all vertex degrees at most k is $(k+1)$-colorable. Consider a particular graph $G = (V, E)$ with $n + 1$ vertices, and all degrees at most k. Let G' be the graph obtained from G by deleting a particular vertex v and all the edges incident to v. That is, G' is the incident subgraph of G on the vertices $V \setminus \{v\}$. G' has n vertices, all of degree at most k, and is thus $(k + 1)$-colorable. Let c' be such a coloring of G'. We extend it to a coloring c of G as follows. For every vertex $u \in G$ such that $u \neq v$ we define $c(u) = c'(u)$. The vertex v has at most k neighbors in G and there is at least one color i among $\{1, 2, \ldots, k + 1\}$ that has not been assigned to any of them. We define $c(u) = i$. This is a $(k + 1)$-coloring, and the proposition follows. □

15.3 Bipartite graphs and matchings

A bipartite graph is a graph that can be partitioned into two parts, such that edges of the graph only go between the parts, but not inside them. Formally, a graph $G = (V, E)$ is said to be bipartite if and only if there exist $U \subseteq V$, such that

$$E \subseteq \{\{u, u'\} : u \in U \text{ and } u' \in V \setminus U\}.$$

The sets U and $V \setminus U$ are called the *classes* of G. A *complete bipartite graph* $K_{m,n}$ is a graph in which all the edges between the two classes are present. Namely, $K_{m,n} = (V, E)$, where $V = \{1, 2, \ldots, m+n\}$ and $E = \{\{i, j\} : 1 \leq i \leq m, m+1 \leq j \leq m+n\}$. The number of edges in K_n is mn. From the definition of coloring, it follows that a graph is bipartite if and only if it is 2-colorable. (Check!) Here is another useful characterization of bipartite graphs:

Proposition 15.3.1. *A graph is bipartite if and only if it contains no cycle of odd length.*

Proof. For one direction of the claim, let G be a bipartite graph and let $C = (v_1, v_2, \ldots, v_n, v_1)$ be a cycle in G. Suppose without loss of generality that $v_1 \in U$, where U is as in the definition of bipartiteness. Then by simple induction that we omit, $v_i \in U$ for every odd $1 \leq i \leq n$. Since $\{v_n, v_1\} \in E$, $v_n \in V \setminus U$ and thus n is even. It follows that the number of edges in C is even.

Before proving the other direction, we need a simple lemma.

Lemma 15.3.2. *Given a graph $G = (V, E)$, let $P = (v_1, v_2, \ldots, v_n)$ be a shortest path between two vertices v_1 and v_n in G. Then for all $1 \leq i < j \leq n$, $P_i = (v_i, v_{i+1}, \ldots, v_j)$ is a shortest path between v_i and v_j.*

Proof. Proof by contradiction. Let $Q_i = (v_i, u_1, u_2, \ldots, u_l, v_j)$ be a shortest path between v_i and v_j. Assume for the sake of contradiction that Q_i is shorter than P_i. Consider the walk

$$Q = (v_1, v_2, \ldots, v_i, u_1, u_2, \ldots, u_l, v_j, v_{j+1}, \ldots, v_n).$$

Since Q_i is shorter than P_i, Q is shorter than P. Now consider the graph $G' = (V(Q), E(Q))$. This graph is connected, and thus there is a shortest path P' between v_1 and v_n in G'. The number of edges in this shortest path cannot exceed the total number of edges in G', and thus P' is shorter than P. Since P' is also a path between v_1 and v_n in G, we have reached a contradiction. □

We now turn to the other direction of the proposition. Assume that $G = (V, E)$ has no odd cycle. If G has more than one connected component we look at every component separately. Clearly, if every component is bipartite, G as a whole is bipartite. Thus assume that G is connected. Pick an arbitrary vertex $v \in V$ and define a set $U \subseteq V$ as

$$U = \{x \in V : \text{ the shortest paths from } v \text{ to } x \text{ have even length}\}.$$

Clearly, $V \setminus U$ is the set

$$V \setminus U = \{x \in V : \text{ the shortest paths from } v \text{ to } x \text{ have odd length}\}.$$

We prove that no two vertices in U are adjacent; the proof for $V \setminus U$ is similar. Consider for the sake of contradiction an edge $e = \{u, u'\} \in E$, such that $u, u' \in U$. Denote a shortest path from v to u by P_1 and a shortest path from v to u' by P_2. Given two vertices s and t on a path P, let $P^{s,t}$ be the part of P that connects s and t. Consider a vertex w that lies on both P_1 and P_2. The above lemma implies that $P_1^{v,w}$ and $P_2^{v,w}$ are shortest paths between v and w and thus have the same length, which we denote by $l(w)$. Consider the vertex w^* shared by P_1 and P_2 that maximizes $l(w)$ among all such w. The paths $P_1^{w^*,u}$ and $P_2^{w^*,u'}$ share no vertex in common other than w^*. Furthermore, the length of $P_1^{w^*,u}$ is the length of P_1 minus $l(w^*)$ and the length of $P_2^{w^*,u'}$ is the length of P_2 minus $l(w^*)$. Since the lengths of P_1 and P_2 are

both even, the lengths of $P_1^{w^*,u}$ and $P_2^{w^*,u'}$ have the same parity (that is, they are either both even or both odd). Now consider the cycle C composed of $P_1^{w^*,u}$, the edge $\{u, u'\}$, and $P_2^{w^*,u'}$. Since $P_1^{w^*,u}$ and $P_2^{w^*,u'}$ share no vertex in common other than w^*, C really is a cycle in G. Moreover, since the lengths of $P_1^{w^*,u}$ and $P_2^{w^*,u'}$ have the same parity, the number of edges in C is odd! We have reached a contradiction that completes the proof. □

Bipartite graphs are particularly useful to model symmetric relations from one set to another. For example, given a collection of boys and girls, we could model the relation "wants to go to the prom with" by a bipartite graph. Given such preferences, an interesting question is whether we can pair the boys up with the girls, so that they all end up going to the prom with someone they actually want to go with. It turns out that this question has a very precise answer. To state the theorem we need to define the notion of matching:

Definition 15.3.3. *Given a bipartite graph $G = (V, E)$, a* matching *B in G is a set of disjoint edges. Namely, $B \subseteq E$ and $e_1 \cap e_2 = \emptyset$ for any $e_1, e_2 \in B$. A matching is said to be* perfect *if $\bigcup_{e \in B} e = V$.*

Consider now a set B of boys, a set G of girls, and a symmetric relation P from B to G. Define a graph $W = (B \cup G,\ \{\{b, g\} : (b, g) \in P\})$. The above question simply asks to characterize when there exists a perfect matching in W. The below result, known as Hall's theorem, provides such a characterization. To state the theorem, we use another piece of notation: Given a subset S of the vertices of W, we let $\Gamma(S)$ be the set of vertices of W adjacent to at least one of the vertices of S.

Theorem 15.3.4. *A bipartite graph $W = (V, E)$ with classes B and G has a perfect matching if and only if $|B| = |G|$ and $|\Gamma(S)| \geq |S|$ for all $S \subseteq B$.*

Proof. One direction is easy: Assume W has a perfect matching and consider a set $S \subseteq B$. Every element of S is matched to a distinct element of G and hence $|\Gamma(S)| \geq |S|$. In particular, $|G| \geq |B|$. By a symmetric argument we get that $|B| \geq |G|$ and thus $|B| = |G|$.

For the other direction, assume that $|B| = |G|$ and that $|\Gamma(S)| \geq |S|$ for all $S \subseteq B$. We prove that there exists a perfect matching in W by strong induction on $|B|$. For the base case, if $|B| = |G| = 1$, the matching consists of the single edge of W. Assuming that the claim holds for all graphs with $|B| \leq k$, consider a graph W as above with $|B| = k + 1$. We distinguish between two possibilities:

(a) If for every $S \subset B$, $|\Gamma(S)| > |S|$, we take an arbitrary $x \in B$ and match it with an adjacent $y \in G$. Then for every subset S' of $B \setminus \{x\}$, it still holds that the number of vertices of $G \setminus \{y\}$ adjacent to at least one of the vertices of S' is at least $|S'|$. We can thus match the vertices of $B \setminus \{x\}$ with the vertices of $G \setminus \{y\}$ by the induction hypothesis.

(b) If for some $S \subset B$, $|\Gamma(S)| = |S|$, we note that for every $S' \subseteq S$, the number of vertices in $\Gamma(S)$ adjacent to at least one of the vertices of S' is at least $|S'|$.

Thus we can match S with $\Gamma(S)$ by the induction hypothesis. Now we need to show that we can match $B \setminus S$ with $G \setminus \Gamma(S)$. Consider a set $S' \subseteq B \setminus S$ and the set T' of its neighbors in $G \setminus \Gamma(S)$. Note that the set of neighbors of $S \cup S'$ in G is $\Gamma(S) \cup T'$. Thus $|S \cup S'| \leq |\Gamma(S) \cup T'|$. Since $|S| = |\Gamma(S)|$, we get that $|S'| \leq |T'|$. Thus by the induction hypothesis we can also match $B \setminus S$ with $G \setminus \Gamma(S)$.

This shows that in both cases all the vertices of B can be matched with vertices of G as required, and concludes the proof. $\square$