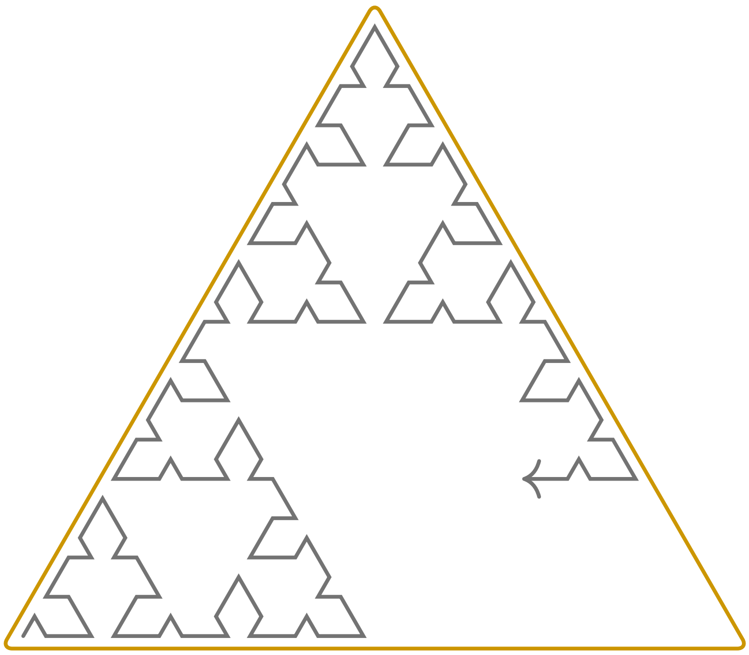
Mikhail Lavrov

Start Doing Graph Theory

Appendix



available online at https://vertex.degree/

Contents

About this document

Where it's from

If you downloaded this PDF file yourself from https://vertex.degree/contents, presumably you know what you were doing. But in case you're confused or got the PDF file somewhere else, let me explain!

This is not an entire graph theory textbook. It is one part of the textbook *Start Doing Graph Theory* by Mikhail Lavrov, in case it's convenient for you to download a few smaller files instead of one large file. You can find the entire book at https://vertex.degree/; all of it can be downloaded for free.

The complete textbook has some features that the individual parts couldn't possibly have. In this PDF, if you click on a chapter reference from a different part of the book, then you will just be taken to this page, because I can't take you to a page that's not in this PDF file. The hyperlinks in the complete book are fully functional; there is also a preface and an index.

What's inside

This is the appendix of *Start Doing Graph Theory*, and includes two chapters on writing proofs. Appendix A has general advice on proof-writing, and Appendix B covers proofs by induction. In both cases, I primarily focus on writing proofs about graphs, and almost all of the examples focus on graph theory.

Depending on your level of mathematical preparation, you may or may not need this appendix, but if you do read it, I assume that you read it after or at the same time as the first part of the textbook: Chapters 1 through 4.

The cover

The cover of this PDF shows most of a Hamilton cycle in a graph representing the Towers of Hanoi problem (with 4 disks).

The license

Appendix

A A review of proof-writing

The purpose of this chapter

While I hope that these little sections at the beginning of each chapter are always useful, the note in this chapter is much more important to read!

It would be far too ambitious of me to hope to write a single chapter that will teach you how to write a proof. Entire books have been written on this subject. Here are a couple that have been made freely available online by their authors:

- *Book of Proof* by Richard Hammack [2].

  You can find it online at `https://richardhammack.github.io/BookOfProof/`.

- *An Infinite Descent into Pure Mathematics* by Clive Newstead [3].

  You can find it online at `https://infinitedescent.xyz/`.

A classic book that should also be mentioned is *How to Solve It* by George Pólya [4]. It is not a book about the method of writing a proof, but will give you advice on how to come up with an idea for it, which is maybe even more valuable.

Usually, when you learn proof-writing, you begin by proving statements that are mathematically easy, so you don't have to struggle with two things at once. After you've done that, you might not yet be comfortable with writing proofs on topics that challenge you mathematically. If this is the level you're at, then this chapter is for you.

I will try to prepare you for the kind of proofs you will encounter in this book by showing you how the general ideas of proof-writing specialize to graph theory. A proof is a proof in every area of math, but there are some ideas that show up in graph theory much more often than in number theory or real analysis, for example. When you get used to proofs in several areas of math, you will be an experienced mathematician. You'll be able to jump into a new topic much more confidently.

So that you can benefit from this chapter early, I'll limit myself to examples from the first part of this book (Chapters 1 through 4), but I will mention chapters later in the book where you'll see more examples.

A.1 Conditional statements

In the moment when you first start trying to prove a theorem, you may not yet know how the proof will go. However, just from the statement of the theorem, you can have a rough overall idea of the strategy you need to take. This is a good way to fight "blank page syndrome", where you're staring at a problem and have no clue what to write!

So what do we look for in a statement? A big part of it is conditional statements: statements that can be expressed in the form "If P, then Q". If you're trying to prove such a statement, you can try:

- A **direct proof**, where you assume that P is true, and try to prove that Q is true.
- A **proof by contrapositive**, where you assume the negation of Q, and try to prove the negation of P. (This is a direct proof of the **contrapositive**, "If not Q, then not P", which is an equivalent form of the original conditional statement.)
- A **proof by contradiction**, where you simultaneously assume both P and the negation of Q, and try to prove any statement you know is false.

A big reason why conditional statements are confusing to beginners is that the informal meaning of "if P, then Q" in the English language is under-specified. If you say a sentence like this in casual conversation, only context can determine what you mean in the case that P is not true. Consider the following examples:

- "If you've lived in Paris your whole life (P), you're French (Q)." Here, Q can still be true even if P is false: many French people have never been to Paris.
- "If you spend at least \$50 (P), you will get free shipping (Q)." Here, the offer implies that you will not get free shipping unless you spend at least \$50: when P is false, Q is also false.
- "If you're hungry (P), there's pizza in the fridge (Q)." Here, the condition is irrelevant, and Q is true regardless of the status of P.

In formal mathematics, an "if P, then Q" statement always goes hand-in-hand with an implicit "and if not P, then anything could be true". It is an assertion that Q is true, but limited only to situations where P is true, without making any claim about situations where P is false. When we want to talk about both situations, and say that when P is false, Q must also be false, the standard formulation is "Q if and only if P".[1]

Even mathematicians are not perfect at maintaining this distinction in all circumstances. One place where you will often see the word "if" misused, from a formal point of view, is in definitions. Though I have avoided it in this book, it is common to see definitions phrased as, for example, "A walk $(x_0, x_1, \ldots, x_l)$ is closed if $x_0 = x_l$." All definitions should be read as if-and-only-if statements: a walk is closed if $x_0 = x_l$, and not otherwise.

Finally, let me say a bit about necessary conditions and sufficient conditions. This is another view of conditional statements. If the statement "If P, then Q" always holds, we say that P is a **sufficient condition** for Q: knowing that P is true is enough to know that Q is true. We call Q a **necessary condition** for P, in the sense that without Q happening, P can't happen

[1] Thus, a mathematician would write, "You will get free shipping if and only if you spend at least \$50."

either: this is a paraphrase of the contrapositive, "If not Q, then not P". We don't always use these terms, though: we use them to emphasize specific ways of thinking about a conditional statement.

We say that P is a sufficient condition for Q to emphasize that we're in a scenario where Q is normally a difficult statement to evaluate, but P is an easy-to-check test that can sometimes tell us that Q is true. For example, Corollary 4.7 is a conditional statement where P is "A graph G with n vertices has at least n edges" and Q is "G contains a cycle". Often, counting vertices and edges is much easier than investigating the graph's structure to see if it has a cycle or not. If we count the vertices and edges and the hypothesis of Corollary 4.7 holds, then we can skip that investigation! The condition we've checked is sufficient to know that the graph has a cycle without looking for it.

We say that Q is a necessary condition for P in the reverse scenario: when P is difficult to investigate, and Q is a simple test. However, with necessary conditions, the test has a different meaning: we learn nothing if we check Q and it is true, but if we check Q and it is false, we learn that P must also be false. For example, Proposition 2.2 tells us that if G and H are isomorphic, then $|V(G)| = |V(H)|$ and $|E(G)| = |E(H)|$. Determining whether two graphs are isomorphic is hard, but counting the vertices and edges is a simple initial test we can do. If $|V(G)| \neq |V(H)|$, or if $|E(G)| \neq |E(H)|$, we can skip the hard work: the necessary condition doesn't hold, so G and H are definitely not isomorphic!

If we have an if-and-only-if relationship between P and Q, then either statement is said to be a **necessary and sufficient condition** for the other. If we discover such a relationship, and one of the statements is a simple test, then we can consider the other statement to be easy to check as well. Knowing whether P is true or false tells us all about Q, and vice versa.

A.2 Quantifiers

Whichever proof strategy you select, it will give you some goals to work toward, and some initial assumptions to work with. Usually, those goals and assumptions will contain quantifiers, which give you further structure. Quantifiers come in two types: existential and universal.

An **existential quantifier** is fancy terminology for a piece of a statement which says that some kind of object exists. The notation "$\exists x \in S, P(x)$" is shorthand for saying, "There exists an object x in the set S for which $P(x)$ is true."

Typically, we prove such a claim by constructing an example; in simple cases, that just means writing down what it is. For example, suppose you want to prove the following statement: "The circulant graph $\mathrm{Ci}_8(3)$ is isomorphic to C_8." Two graphs G and H are isomorphic if there exists an isomorphism between them: a function $\varphi\colon V(G) \to V(H)$ with certain properties. Your proof might begin by defining a function φ, and then checking that it has the properties that make it an isomorphism between $\mathrm{Ci}_8(3)$ and C_8.

I should also describe what happens when an existential statement is part of your assumptions. In that case, you get to summon up an object of the type being described, and start using it in your proof; this can be incredibly helpful! For example, suppose you are giving a direct proof of the following statement: "If a graph G has an $x - y$ walk, then it has an $x - y$ path." The existence of an $x - y$ walk is one of your assumptions: you get to assume that such a walk exists.

It's often a good idea to describe the object in detail, giving names to its moving parts; you might begin by saying, "Let the sequence $(x_0, x_1, \dots, x_l)$ be an $x - y$ walk, with $x_0 = x$ and $x_l = y$." Later on in the proof, you will get to manipulate the vertices $x_0, x_1, \dots, x_l$, and use facts about them that are based on the definition of an $x - y$ walk.

Moving on, a **universal quantifier** says that a statement is always true: it is a universal rule. The notation "$\forall x \in S, P(x)$" is shorthand for saying, "For all objects x in the set S, the statement $P(x)$ is true."

Question: What if the set S is the empty set?

Answer: In that case, a statement about all elements of S is automatically true, because there's nothing to check: we say it is **vacuously true**.

We don't generally make such statements on purpose, but we might get them as a special case of a more general theorem.

To prove such a statement, we need an argument that applies to every element of S at once. There is a standard way to phrase such an argument: it is to choose an arbitrary element $x \in S$, assuming nothing else about it. Then, your goal is to prove that $P(x)$ is true. This can be easy or hard, but look on the bright side: in this scenario, you both get to start with an element $x \in S$ to work with, and get an idea of how you want your proof to end!

For example, suppose you want to prove the following statement: "For all integers $n \geq 3$, the cycle graph C_n is connected." You would begin by taking an integer $n \geq 3$, and writing the rest of your proof for that value of n. As long as you don't accidentally sneak in any further assumptions about n, your argument will apply to every integer you could have chosen.

The case I find most frustrating is the case of assuming a "for all" statement, because you can't do anything with it when you begin. For example, suppose you have a graph G for which you've made the assumption, "All cycles in G have an even length." (See Chapter 13 to learn more about such graphs.) You can't do anything right away—it's only once you encounter a cycle in G that this assumption "triggers" and tells you that the length of the cycle is even. It's possible that G has no cycles, in which case you will never get to use the assumption.

To give you an overview of the situation at a glance, Figure A.1 describes what happens to both types of quantifiers in both cases: when you assume them, and when you prove them. Since some cases are easier to deal with than others, you can try to change which case you have to deal with by changing your approach.

- You could try a different proof method that changes what you have to do: for example, instead of proving a statement, you might assume its negation.
- You could use a theorem which gives an alternate characterization of an object, with a different type of quantifier.

	When you get to assume it:	**When you have to prove it:**
$\forall x \in S, P(x)$	Nothing can be done right away. Whenever you encounter elements of S, then you can assume that P is true for those elements.	Tells you how to begin and end. Begin by defining an element $x \in S$; you can't control anything else about x. Your goal is to prove $P(x)$.
$\exists x \in S, P(x)$	Tells you how to begin the proof. Begin by defining an element $x \in S$ and assuming that $P(x)$ is true; you can't control anything else about x.	Tells you how to end the proof. You must somehow construct an element $x \in S$ (you don't get to start with one) for which $P(x)$ is true.

Figure A.1: How to use quantifiers in a proof

In more complicated statements, you will have to juggle both kinds of quantifiers at once. An especially important combination to look at is the combination $\forall x, \exists y$: "For every x, there exists a y." This is a very demanding statement to prove, because you must present an example of y, which might have to depend on x; however, you can't make any assumptions about x. Let me give you some examples to compare:

- "The circulant graph $\mathrm{Ci}_8(3)$ is isomorphic to C_8" is a concrete, purely existential statement: no universal quantifiers at all. You can prove that an isomorphism exists by writing down what it is and checking the conditions.
- "For all odd $n \geq 3$, the circulant graph $\mathrm{Ci}_n(2)$ is isomorphic to C_n" has a universal quantifier on n, but an existential quantifier hidden inside the word "isomorphic". You might still be able to write down an isomorphism, but you will have to write it down as a formula in terms of n.
- "Every connected n-vertex graph G in which all vertices have degree 2 is isomorphic to C_n" is even trickier: it has a universal quantifier on a graph G, which is a much more complicated object than a number!

  You will not be able to write down an isomorphism, not even with the aid of formulas, because you can't plug a graph G into a formula. In your proof, you might describe a general strategy for finding an isomorphism, and check that it always works.

This situation is part of the reason why algorithms play a big role in graph theory: a common way to prove that something exists is to give an algorithm for finding it. Theorem 8.3 and Theorem 8.4 are good examples of this technique early in the book.

I should also warn you that often, universal quantifiers are omitted in the statement of a theorem. If the statement of a theorem contains variables that don't have a quantifier attached, this usually means that the theorem should be true for all possible values of those variables; the scope should hopefully be clear from context.

I try to avoid doing this too much, but consider for example Corollary 4.7, which I guess I shouldn't go back and edit because then I won't be able to use it as an example here. The full statement of the corollary is: "If G has n vertices and at least n edges, then G contains a cycle." Neither G nor n is quantified, so we interpret both of them as having hidden universal quantifiers on them. The statement should be true for all graphs G and for all integers $n \geq 1$.

> **Question:** How do we know what kind of variables G and n are?
>
> **Answer:** Part of this is convention: G usually refers to a graph and n usually refers to an integer. Part of this is context: G is mentioned as having vertices and edges, so it should be a graph, and n is the number of vertices in G, so it can only be a positive integer.

A.3 Unpacking definitions

Quantifiers, implications, and many other parts of a statement can be tucked away inside a definition where you can't easily see them. For example, it is not obvious from reading "G and H are isomorphic" that it's existential ($\exists$) statement: that there exists a function $\varphi\colon V(G) \to V(H)$ which is an isomorphism. (Inside the definition of an isomorphism, more quantifiers are tucked away!)

> **Question:** What are the quantifiers in "G is connected?"
>
> **Answer:** It's a $\forall\exists$ statement: for every two vertices $x, y \in V(G)$, there exists an $x - y$ walk.

> **Question:** What are the quantifiers in "The maximum degree of G is k?"
>
> **Answer:** It has two parts: a $\forall$ statement that every vertex has degree at most k, and an $\exists$ statement that there exists a vertex of degree k. (More on such statements later!)

This means that there's an important first step you have to keep in mind whenever you write a proof: you must unpack all the definitions you're working with to get at the moving parts inside them!

Let's look at an example of this: the proof of Lemma 3.3. This lemma claims that the relation $\leftrightsquigarrow$ on the vertices of a graph G is an equivalence relation, where $x \leftrightsquigarrow y$ is defined to mean that there is an $x - y$ walk in G.

To begin with, we are proving that something is an equivalence relation. By definition, an equivalence relation is a relation which is reflexive, symmetric, and transitive. So right away, we know that our proof will have three parts, one where we prove each part of the definition.

Let's look at just one of these: proving that $\leftrightsquigarrow$ is symmetric. The definition of this is that for all x and y (which, in this case, are vertices of G), if $x \leftrightsquigarrow y$, then $y \leftrightsquigarrow x$.

Question:	Assuming that we choose to write a direct proof (which there's no reason not to do), which quadrant of Figure A.1 are we in?
Answer:	The top right quadrant: we are proving a universal statement about all pairs of vertices $x, y \in V(G)$.

Therefore we begin by choosing arbitrary vertices $x, y \in V(G)$. We assume that $x \leftrightsquigarrow y$ is true, and set ourselves a goal: to prove that $y \leftrightsquigarrow x$ is true. At this point, we must also unpack the definition of $x \leftrightsquigarrow y$. We are assuming that there is an $x - y$ walk in G; we are proving that there is a $y - x$ walk in G.

Question:	Which quadrants of Figure A.1 do these fall under?
Answer:	The bottom two quadrants: both the assumption we make about x and y and the property we want to prove are existential statements. We get to summon an $x - y$ walk out of nowhere, but we will have to construct a $y - x$ walk.

It is not very helpful to use the existence assumption merely by saying something like, "Let W be an $x - y$ walk in G." To get anything useful out of the assumption, we should unpack the definition of an $x - y$ walk. We say, "Let the sequence $(x_0, x_1, \dots, x_l)$ be an $x - y$ walk, with $x_0 = x$ and $x_l = y$."

Looking ahead, we see that we'll want to define a new sequence of vertices, which we will then prove is a $y - x$ walk. It is only at this point that we get to the problem-solving step of the proof! To figure out how to get a $y - x$ walk out of an $x - y$ walk, we might look at some small examples to get a feeling for the problem. With experience comes an "intuition" for things to try, which is just a way to say that you've seen similar problems before and have some guesses about what might work.

To summarize, here is the "scaffolding" of a proof that $\leftrightsquigarrow$ is symmetric; only the sections with "…" are left for us to figure out. (Not every proof is as heavy in definitions, and so you won't always have as much unpacking to do.)

Proof. Let x and y be two arbitrary vertices, and suppose that $x \leftrightsquigarrow y$: that there is a $x - y$ walk in G. Our goal is to prove that there is also a $y - x$ walk in G.

Let $(x_0, x_1, \dots, x_l)$ be an $x - y$ walk (with $x_0 = x$ and $x_l = y$). Then define a new sequence of vertices by …

This sequence of vertices starts at y, ends at x, and consecutive vertices in the sequence are adjacent because …, proving that it is a $y - x$ walk.

Since a $y - x$ walk exists, we conclude that $y \leftrightsquigarrow x$. Since this is true for all pairs of vertices $x, y \in V(G)$, we conclude that that $\leftrightsquigarrow$ is symmetric. □

I should also mention that there is a danger to unpacking too far. If you unpack every definition you can, you are forced to write a "low-level" proof that works with the fundamental notions of graph theory. The alternative is a "high-level" proof, where you stop early to apply a theorem you know to an object without unpacking it.

For example, in Chapter 4, we gave Theorem 4.4 a low-level proof: we proved that a cycle exists by listing out the vertices of a walk representing it. Later, we gave Corollary 4.7 a more high-level proof: we proved that a cycle exists by applying Theorem 4.4, without interacting directly with the definition of a cycle.

A.4 Optimization problems

Many definitions in graph theory involve solving an optimization problem: they are phrased in terms of the biggest or smallest thing of a certain type. In the first part of the textbook, this includes:

- The distance between two vertices x and y: the length of the shortest $x - y$ walk.
- The minimum and maximum degree of a graph G: the smallest and largest, respectively, of the degrees of the vertices of G.

When we say, "The distance between vertices x and y is 5," we are making two separate claims. First, we are saying that there is in fact an $x - y$ walk of length 5. Second, we are saying that this is the shortest walk: all $x - y$ walks have length at least 5. The first claim is an existential ($\exists$) statement, and the second claim is a universal ($\forall$) statement.

Question: What statement about the distance $d(x, y)$ do we make if we only say that there is an $x - y$ walk of length 5?

Answer: This is equivalent to saying that $d(x, y) \leq 5$. The distance can't be longer than 5, because we've already found walk of length 5, but it might be shorter.

Question: What statement about the distance $d(x, y)$ do we make if we only say that that all $x - y$ walks have length at least 5?

Answer: This is equivalent to saying that $d(x, y) \geq 5$. We can't do better than 5, but we don't know if we can actually achieve 5.

What we see for proofs about the distance between two vertices is true in general. An upper bound on a minimization problem is an existential statement: we can prove it by giving a single example of a solution that achieves that bound. A lower bound on a minimization problem is a universal statement: we can prove it by proving that no solution can do better. For maximization problems, the situation is reversed.

Let's look at an example.

Proposition A.1. *Let G be the graph with vertex set $V(G) = \{1, 2, \dots, 100\}$ in which two vertices x and y are adjacent if and only if $|x - y|$ is the square of a positive integer. (For example, vertices 7 and 71 are adjacent, because $|7 - 71| = 64 = 8^2$.)*

Then G has maximum degree $\Delta(G) = 14$.

Proof. First, we prove that a vertex of degree 14 exists, by example. The vertex 50 is adjacent to the 14 vertices

$$1, 14, 25, 34, 41, 46, 49, 51, 54, 59, 66, 75, 86, 99.$$

We don't have to check them all by hand: these are the vertices $50 - k^2$ for $k = 1, 2, \dots, 7$ and $50 + k^2$ for $k = 1, 2, \dots, 7$. All we have to do is to check that the extreme values $50 - 7^2 = 1$ and $50 + 7^2 = 99$ still fall within $V(G)$.

This example proves that $\Delta(G) \geq 14$; to prove that $\Delta(G) \leq 14$, we prove that all vertices have degree at most 14.

Let x be an arbitrary vertex. From the example we looked at, we can see that to find the degree of x, we need to count how many values of the form $x - k^2$ or $x + k^2$ fall with in the range $\{1, 2, \dots, 100\}$. Let a be the largest integer such that $x - a^2 \geq 1$, and let b be the largest integer such that $x + b^2 \leq 100$; then the neighbors of x are $x - 1^2, x - 2^2, \dots, x - a^2$ and $x + 1^2, x + 2^2, \dots, x + b^2$, and x has degree $a + b$.

> **Question:** What is the largest possible value of a?
>
> **Answer:** It is 9: even if x is as large as possible, that is if $x = 100$, then $x - 10^2$ is still not in the range $\{1, 2, \dots, 100\}$.

Similarly, we can prove that the largest possible value of b is 9, giving us an upper bound of $9 + 9 = 18$ on the degree of x, and on the maximum degree. But that's not good enough; we wanted $\Delta(G) \leq 14$, not $\Delta(G) \leq 18$.

> **Question:** What prevents us from making both a and b this large simultaneously?
>
> **Answer:** To get a to reach 9, we need x to be close to the end of the range. To get b to reach 9, we need x to be close to the start of the range.

We know that having $a = 7$ and $b = 7$ are possible when $x = 50$, so let's rule out the remaining possibilities. If $a \in \{8, 9\}$, then $x - 8^2 \geq 1$, so x is at least $1 + 8^2 = 65$; but now, since $65 + 6^2 = 101$ is too big, we learn that $b \leq 5$. Similarly, if $b \in \{8, 9\}$, then $x + 8^2 \leq 100$, so x is at most $100 - 8^2 = 36$; but now, since $36 - 6^2 = 0$ is too small, we learn that $a \leq 5$. So in these cases, no better solution than $a + b = 9 + 5 = 14$ or $a + b = 5 + 9 = 14$ is possible.

> **Question:** Is $(a, b) = (9, 5)$ or $(a, b) = (5, 9)$ actually achievable?
>
> **Answer:** No: for example, $a = 9$ requires $x \geq 82$, and $b = 5$ requires $x \leq 75$. This is fine for our proof: we are done with the stage that proves existence, now we are only ruling out options, and we don't care about ruling out options that don't contradict the upper bound $\Delta(G) \leq 14$ we want.

When $a \in \{8, 9\}$ or $b \in \{8, 9\}$, we have $\deg(x) = a + b \leq 14$. But if $a \leq 7$ and $b \leq 7$, we also have $\deg(x) = a + b \leq 14$. Therefore $\deg(x) \leq 14$ no matter what x is. This tells us that $\Delta(G) \leq 14$, which completes the overall proof. □

In Chapter 5, the diameter of a graph G is introduced: the largest distance between two vertices of G. The definition of diameter has one optimization problem nested inside another! This means that both lower and upper bounds on the diameter of a graph unpack to statements with multiple quantifiers.

For example, suppose we want to prove that the diameter of G is at most k. Then we want to prove that $d(x, y) \leq k$ for all pairs of vertices x, y, which is a $\forall\exists$-type statement: for all x and y, there exists an $x - y$ walk of length at most k.

Question:	What if we try to prove that the diameter of G is at least 5?
Answer:	This will be an $\exists\forall$-type statement: we want to prove that there exist vertices x and y such that all $x - y$ walks have length at least 5.

Here are some of the notable optimization problems introduced later in the book:

- The maximum matching problem and the minimum vertex cover problem, introduced in Chapter 13.
- The independence number $\alpha(G)$ and clique number $\omega(G)$, defined in Chapter 18.
- The vertex and edge coloring problems studied in Chapter 19 and Chapter 20.
- The connectivity $\kappa(G)$ and several related parameters, defined in Chapter 26.

A.5 Algorithms

Algorithms play a role in graph theory for two reasons.

First of all, there are many applications of graph theory in software, so computer scientists think about graph algorithms.

Second, even pure mathematicians that don't care about computer science at all can be interested in graph algorithms as an aid in proof-writing. One way to prove that an object exists is to give an algorithm for finding it.

In both cases, but especially in the second case, it's important for the algorithm to be accompanied by a proof of correctness. In most respects, this is a proof like any other. There are a few proof techniques that are unusually common when dealing with algorithms, which we'll see in a moment. Before that, there's one feature of the proof that I want to point out, because it's easy to miss if you've never thought about it before. It is important to prove that the algorithm eventually stops, and doesn't just keep iterating forever!

In the first part of the book, there is only one notable algorithm: breadth-first search, which we use to compute distances in a graph at the end of Chapter 3. Breadth-first search continues to be useful in algorithms throughout this book: as late as Chapter 28, we return to it to find

(a) A cycle in the cube graph (b) Replacing the cycle

Figure A.2: An example of the operation in Problem A.1

a shortest path in a residual network. In this algorithm, the stopping condition is relatively straightforward. In each step, we either explore at least one new vertex, or we stop because we didn't find any new vertices. We cannot keep exploring new vertices forever: eventually, we run out of vertices in the graph to explore!

Such situations are common, but not universal. In this textbook, the most sophisticated proof that an algorithm stops is Theorem 28.7, for an algorithm to find maximum flows. In that case, the algorithm deals with real numbers, and it's possible to keep increasing a real number forever even if there's an upper bound on the number; this is what makes the analysis tricky!

To see a tricky proof that an algorithm stops that only needs topics from the first part of the book, let me give an example from the 2019 Princeton University Mathematics Competition (PUMaC): problem 1 from the individual final round in Division A [5]. The problem does not talk about algorithms, but it describes an iterated procedure performed on a graph, so many of the same ideas come up. Here it is:

Problem A.1. *Given a connected*[2] *graph G and cycle C in it, we can perform the following operation: add another vertex v to the graph, connect it to all vertices in C and erase all the edges from C. Prove that we cannot perform the operation indefinitely on a given graph.*

An example of this operation is shown in Figure A.2. Since the graph gets bigger and bigger every time we perform the operation, it's not at all obvious that we'll eventually have to stop!

To solve the problem, we'll need two lemmas.

Lemma A.2. *After the operation in Problem A.1 is performed on a connected graph, the result is another connected graph.*

Proof. Let G be the graph we started with and let H be the result of performing the operation.

For any two vertices $x, y \in V(G)$, there is an $x - y$ walk in G, because G is connected. We can turn this $x - y$ walk in G into an $x - y$ walk in H, even though not all edges of G are still present in H. Every time that the walk goes from a vertex u to a vertex w along an edge of C, replace that step by two steps, u to v to w.

[2]The problem statement on the PUMaC website does not say that G is connected, but the official solution assumes a connected graph. If G is not connected, the same argument can be applied to each connected component. I've decided to add the assumption just so that we can skip this step.

This proves that all vertices of $V(G)$ are in the same connected component of H. In H, there is one more vertex: the vertex v we added. It is in the same component as the other vertices, because it is adjacent to the vertices on C; this proves that H is connected. □

Lemma A.2 is a good example of proving an **invariant** of an algorithm (not to be confused with a graph invariant). By showing that some property (in this case, the connectedness of the graph) does not change after we perform the operation once, we can conclude that it never changes. In this case, it follows from Lemma A.2 that no matter how many times we perform the operation in Problem A.1, we will have a connected graph.

We can obtain another invariant by counting the vertices and edges.

Question:	In Figure A.2, what is the number of vertices before and after the operation, and what is the number of edges?
Answer:	The number of vertices goes from 8 to 9. The number of edges remains at 12: we deleted 4 edges on the cycle, but added 4 edges from the vertices of the cycle to the new vertex.

This example is probably enough to guess what happens in general:

Lemma A.3. *After the operation in Problem A.1 is performed on a graph with n vertices and m edges, the result is a graph with $n+1$ vertices and m edges.*

Proof. The new graph has $n+1$ vertices because 1 vertex is added and none are removed.

Let the cycle C used in the operation have length k; then C has k vertices and k edges. Deleting the edges of C reduces the number of edges from m to $m-k$, but adding an edge from v to each vertex of C increases the number of edges from $m-k$ back to m. □

Technically, the number of vertices is not an invariant of the algorithm, but a monovariant: rather than always staying the same, it increases in a predictable way.

Armed with these two lemmas, we can explain why the operation can't be continued forever.

Question:	Suppose that we could perform the operation 100 times on the cube graph. What can you say about the result?
Answer:	The result must be a connected graph (by Lemma A.2) with 108 vertices and 12 edges (by Lemma A.3).

Question:	Why is this ridiculous?
Answer:	In a graph with 108 vertices and 12 edges, there are not even enough edges to give every vertex a neighbor! The graph must have many different connected components which are isolated vertices; it certainly cannot be connected.

Generalizing this logic, suppose that we start with a connected graph G that has n vertices and m edges. Assume for the sake of contradiction that we can perform the operation at least $2m$ times. When this is done, we have a connected graph G (by Lemma A.2) with $2m+n$ vertices and m edges (by Lemma A.3).

To see the problem clearly, apply the handshake lemma (Lemma 4.1). If every vertex in G has degree at least 1, then the sum of all $2m+n$ degrees is at least $2m+n$, but the handshake lemma says that the sum of degrees is only $2m$. Therefore G must have some vertices of degree 0. Each such vertex is a connected component all by itself: somehow, G is not connected! This is a contradiction, so it must be impossible to perform the operation $2m$ times.

In fact, using techniques from Chapter 10, we can determine the exact number of times we can perform the operation before we have to stop. When we are forced to stop, we must have a graph that is connected but has no cycles: by Theorem 10.2, this graph must be a tree! A tree with m edges has $m+1$ vertices, so if we started with a connected graph that has m edges and n vertices, we will be able to perform the operation $m-n+1$ times: no more, and no less.

A.6 The extremal principle

Not all proofs of existence are algorithms that tell you how to construct the object we want. There are many proof techniques that let us prove something exists without giving us any clues about how to find one. (For example, in Theorem 18.5, we prove that a graph with a certain property exists by showing that a randomly chosen graph has a positive probability of having that property.)

A notable proof technique of this type is the extremal principle. In the first part of the book, there are two examples of its use:

- In Theorem 3.1, to prove that an $x-y$ path exists, we chose a shortest $x-y$ walk, and then proved that it represents a path.
- In Theorem 4.4, to prove that a cycle exists, we chose a longest path, and then proved that an initial segment of that path can be turned into a cycle.

In general, the **extremal principle** is a way we can make use of an assumption that something exists (as in the bottom left quadrant of Figure A.1) and strengthen that assumption for free. Instead of taking an arbitrary object of that type, we take an object which is as good as possible by some measure.

There are two caveats to this. First of all, you have to know that there are objects of that type. When we chose a shortest $x-y$ walk in the proof of Theorem 3.1, for example, the existence of $x-y$ walks was one of our assumptions. If it was not, then the proof would not be valid: we can't take a shortest $x-y$ walk if there's a possibility that x and y are not in the same connected component.

Question:	When we chose a longest path in the proof of Theorem 4.4, how did we know that a path exists?
Answer:	If nothing else, a graph with at least one vertex should have a path of length 0. (With the assumption that the graph has minimum degree 2, we can even guarantee slightly longer paths than this!)

Second, it must be guaranteed that there is a best object, however you decide to judge "best". Ties are okay, but when there are infinitely many options to choose from, it's possible that no matter which object you choose, there's a better one. In graph theory, we are usually dealing with non-negative integers. In that case, it's always okay to pick the smallest value (as in the case of a shortest $x - y$ walk). In situations where there's an upper bound on the value, we can also pick the largest; however, that's not valid in general.

Question:	Why is there guaranteed to be a longest path in any graph?
Answer:	In a graph with n vertices, a path can also contain at most n vertices, which means that it can have length at most $n - 1$. (There's not necessarily a path of length $n - 1$; this is just an upper bound.)

When should you consider using the extremal principle? Although you could try using it at any time, I've found that there's a specific kind of situation where it's convenient to use: when you can imagine doing something over and over again, the extremal principle can let you "skip to the end" of that process. That's vague, so let me explain how it works in the two examples of this section.

To prove Theorem 3.1, you can imagine taking an $x - y$ walk and cleaning it up to make it represent an $x - y$ path, step by step. How can we do that? Well, an "un-path-like" thing for a walk to do is to revisit a vertex z. Whenever the walk does that, we can eliminate a visit to z by skipping the segment of the walk between the first visit to z and the last. Every time we do this, it makes the walk shorter. Therefore if we skip ahead to an $x - y$ walk that is as short as possible, it must not be possible to do this again.

To prove Theorem 4.4, you can imagine walking around the graph arbitrarily (but without backtracking along the same edge) until you come back to a vertex. When you've visited a vertex for a second time, the trajectory from your first to your second visit must form a cycle. How can we skip ahead to the end here? Well, every time we don't come back to a vertex, we end up making a longer and longer path. So if we skip ahead to the longest path in the graph, we'll be forced to come back to a vertex in the very next step.

A.7 Practice problems

1. Rewrite each of the following statements to make the quantifiers and logical implications inside it explicit.

   a) Two isomorphic graphs always have the same number of edges.

   b) Every n-vertex graph in which all vertices have degree at least $\frac{n-1}{2}$ is connected.

c) Every graph with more edges than vertices has a vertex of degree at least 3.

d) The difference between minimum and maximum degree in a graph can be arbitrarily large.

2. Find the logical negation of each of the following statements, simplifying as much as possible so that no negations of complicated clauses are left. (It is okay if you are left with elementary negations like "x is not adjacent to y".)

   a) Every k-vertex subgraph of G has at most k edges.

   b) There is a set of k vertices in G with no edges between them.

   c) There is a vertex in G adjacent to all other vertices.

   d) For all $n \geq 1$, if an n-vertex graph contains no copies of G, then it can have at most $\frac{1}{3}n^2$ edges.

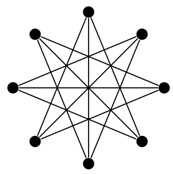
   e) For every vertex x of G, there is another vertex y such that every $x - y$ walk in G has length at least 10.

   f) G contains two vertices x and y such that for every vertex z other than x or y, if z is adjacent to x, then z is also adjacent to y.

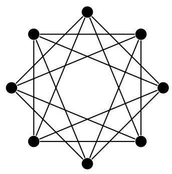
   g) Every graph theorist has a friend that knows somebody the graph theorist doesn't know.

   h) Every textbook has a practice problem that cannot be solved.

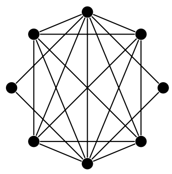
3. Use the three graphs below to answer the questions that follow.



Graph I



Graph II



Graph III

   a) If G is a planar graph with $n \geq 3$ vertices, we know that G has at most $3n - 6$ edges. Using this condition and no other properties of planar graphs, what can we say about graphs I, II, and III?

   b) If G is a graph with $n \geq 3$ vertices and minimum degree at least $n/2$, we know that G is Hamiltonian. Using this condition and no other properties of Hamiltonian graphs, what can we say about graphs I, II, and III?

   c) A connected graph G is Eulerian if and only if every vertex has an even degree. Using this condition and no other properties of Eulerian graphs, what can we say about graphs I, II, and III?

4. a) Prove that the circulant graph $\mathrm{Ci}_8(3)$ is isomorphic to the cycle graph C_8.

   b) Prove that for all odd $n \geq 3$, the circulant graph $\mathrm{Ci}_n(2)$ is isomorphic to C_n.

   c) Prove that for even $n \geq 4$, the circulant graph $\mathrm{Ci}_n(2)$ is not isomorphic to C_n.

5. Find the error in this proof of the statement "Graphs never have edges".

   Let G be any graph, and let x be an arbitrary vertex. Let $(x, x_1, x_2, \ldots, x_l)$ be the longest walk beginning at x.

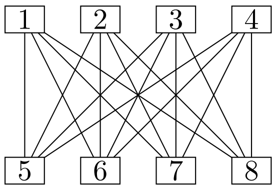
   If there is an edge xy incident to x, then the walk $(x, y, x, x_1, x_2, \ldots, x_l)$ is a longer walk beginning at x. That's a contradiction, because we assumed that we took the longest such walk. So the assumption that the edge xy exists is incorrect: there are no edges incident to x. In other words, x has degree 0.

   Since x was an arbitrary vertex, all vertices of x have degree 0: in other words, the graph G has no edges at all!

6. Write down a scaffolding for a direct proof of the claim "If a graph G is isomorphic to its complement $\overline{G}$, then it is connected."

   That is, unpack all the definitions in the claim and identify all the quantifiers and logical implications. Then, write down all the assumptions and initial definitions you should make, as well as the conclusions you should eventually draw. You don't have to fill in the details of how to get from the start to the end.

7. Prove that the graph shown below has diameter 2:



8. Starting from the complete graph K_{10}, you repeatedly perform the following operation: select 4 vertices of the graph, and toggle the presence of all 6 edges between them (removing them if they are absent, and adding them if they are present).

   Can you ever end up removing all edges of the graph?

9. A P_3-free graph is a graph that does not have the path graph P_3 as an induced subgraph. In other words, if you pick 3 vertices inside a P_3-free graph, there will never be exactly 2 edges between them: there could be 0 edges, a single edge, or all 3 edges.

   a) Let G be a P_3-free graph, and let $\sim$ be the adjacency relation in G: $x \sim y$ if and only if $xy \in E(G)$.

      Prove that $\sim$ is an equivalence relation on $V(G)$.

   b) What does this tell us about the structure of G? Describe what a connected component of G can look like.

B Proof by induction

The purpose of this chapter

Induction is a key proof technique for all mathematicians, but it is especially powerful—and especially complicated!—in graph theory. So this chapter has two reasons to exist.

First, it is a review of induction; once again, I am imagining that some of my readers might be studying graph theory shortly after their first introduction to rigorous proofs. If you've mostly studied induction in the context of proving a closed form for a recurrence relation or a summation, then going to more general proofs by induction is a big step. I want to help you along by showing you plenty of examples, and by doing my best to help make the logic of induction make sense.

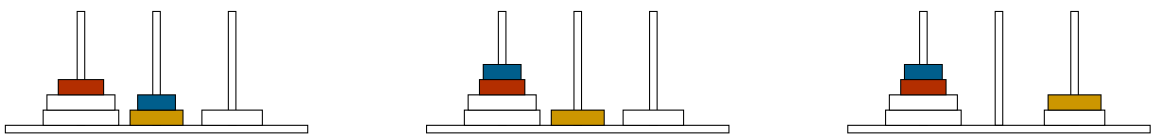
Second, this chapter covers some of the weird things induction can do in graph theory. I especially want to make sure you do not make the mistake in the false Claim B.6, and that you know to use the induction template of "Theorem" B.7 instead, when necessary. I also want to help you discover when induction is a good idea, by describing the kinds of definitions that lend themselves to induction.

These ideas can be applied in other areas of math as well, but because graph theorists study finite objects with not too much structure, it is much more common to be able to tackle a problem by induction.

B.1 The logic of induction

For our first example, let's return to the Towers of Hanoi puzzle, discussed in Chapter 1 and illustrated in Figure B.1. There are three pegs and some number of disks of different sizes stacked on the pegs. In a single move, the top disk on a peg can be moved to the top of a different peg.

The disks stacked on a peg must always form a pyramid, with their sizes increasing from top to bottom. This means that the smallest disk (the one moved from Figure B.1a to Figure B.1b) can always be moved anywhere, but the larger disks are restricted further. From Figure B.1b



(a) A mid-game state (b) The next move (c) Another move

Figure B.1: Two moves in the Towers of Hanoi puzzle to Figure B.1c, we move a disk from the middle peg to the right peg; it cannot be moved to the lft peg instead, because then it would be on top of a smaller disk.

Initially, the disks are all placed on one peg (forming a pyramid, as they always must). The goal of the puzzle is to move all the disks from one peg to another. Our first task in this chapter will be to prove that this puzzle always has a solution.

Theorem B.1. *No matter how many disks there are, it is possible to solve the Towers of Hanoi puzzle, starting with all the disks stacked in a pyramid on any peg, and ending with all the disks stacked in a pyramid on any other peg.*

If you try to solve this puzzle by moving disks at random, you will not get very far. (I can confirm this experimentally. For the six-disk puzzle in Figure B.1, I performed 1000 computer-simulated attempts to solve the puzzle by randomly chosen valid moves. The median number of moves required to move all six disks onto a peg other than the starting one was 8488.) The reason for this is that it's very rare to see a state where one of the larger disks can be moved.

Question: If the largest disk in an n-disk Towers of Hanoi puzzle can be moved from the starting peg, how must the disks be arranged?

Answer: First of all, none of them can be on top of the largest disk. Second, the destination peg to which we want to move the largest disk must also be empty, because the largest disk can't be placed on top of any other. Therefore all $n-1$ other disks must be stacked onto a third peg.

In order to achieve this state, we must move the top $n-1$ disks from one peg to another. In this process, we can ignore the largest disk, which means that we are effectively solving a Towers of Hanoi puzzle with one fewer disk. This makes the problem a perfect setup for an induction proof!

Proof of Theorem B.1. We induct on n, the number of disks. When $n = 1$, we can move the disk from one peg to another in a single step. So the base case holds.

Assume that there is a way to move $n-1$ disks from any peg to any other. Then there is also a solution to the n-disk puzzle, from any starting peg to any ending peg.

1. Using the $(n-1)$-disk solution, move the first $n-1$ disks from the starting peg to the third peg (which is neither the starting nor the ending peg).
2. Move the largest disk from the starting peg to the ending peg.
3. Using the $(n-1)$-disk solution again, move the first $n-1$ disks from the third peg to the ending peg.

By induction, there is a solution for all n. □

This example demonstrates the structure of a proof by induction. Although there are more complicated variants that bend or break some of the rules, an ordinary proof by induction has the following characteristics:

- We must be trying to prove a statement with many cases which can be numbered, usually starting at 0 or 1.

  The first sentence, "We induct on n, the number of disks," is a standard way to explain what the cases are (they are the different versions of the puzzle with different numbers of disks) and how they can be numbered (according to the number of disks, which we are now going to refer to as n).

- We must begin with a **base case**: a self-contained proof of the first case of the statement.

  Here, that's the $n = 1$ case, so at the beginning of the proof, we explain how to solve the 1-disk problem. As in this example, usually the proof of the base case is very short.

- The other part of the proof is an **induction step**: a proof that if any given case of the theorem is true, then the next case is also true.

  As usual, a direct proof of an "if P, then Q" statement begins by assuming P, and trying to prove Q. So we begin the induction step by assuming that the theorem holds in some case, and using this assumption to prove that the theorem also holds in the next case. We refer to this assumption as the **induction hypothesis**.

  The induction step can be written in several slightly different ways. We can assume the n^{th} case of the theorem and prove the $(n+1)^{\text{th}}$, or assume the $(n-1)^{\text{th}}$ case of the theorem and prove the n^{th} case. We can introduce a new variable, proving that if the theorem holds when $n = k$, it also holds when $n = k + 1$ (or that if the theorem holds when $n = k - 1$, it also holds when $n = k$). The choice between these is mostly a matter of taste.

I suspect that many of my readers have already seen proofs by induction, but I am carefully explaining it anyway, because it's a very tricky idea when you first learn it.

It's also often taught without explaining how or why it works. To give you an intuitive understanding, let's prove a few cases of Theorem B.1 without using induction. (In all of these lemmas, by "the puzzle has a solution", we mean that we can move all the disks from any peg to any other peg.)

Lemma B.2. *The Towers of Hanoi puzzle with* 1 *disk has a solution.*

Proof. When no other disks interfere, we can move the disk from any peg to any other in a single step. □

Lemma B.3. *The Towers of Hanoi puzzle with* 2 *disks has a solution.*

Proof. By Lemma B.2, there is a way to move the smallest disk to a different peg. Using this method, move the smallest disk from the starting peg to a third peg: neither the starting nor the ending peg. (In this process, the largest disk won't get in the way, because any other disks can be placed on top of it.)

Now the largest disk is free, and the ending peg is empty, so the largest disk can be moved to the ending peg.

Finally, using the method of Lemma B.2 again, we can move the smallest disk from the peg it's on to the ending peg, which leads to the arrangement we wanted. □

Lemma B.4. *The Towers of Hanoi puzzle with* 3 *disks has a solution.*

Proof. By Lemma B.3, there is a way to move the 2 smallest disks to a different peg. Using this method, move the 2 smallest disks from the starting peg to a third peg: neither the starting nor the ending peg. (In this process, the largest disk won't get in the way, because any other disks can be placed on top of it.)

Now the largest disk is free, and the ending peg is empty, so the largest disk can be moved to the ending peg.

Finally, using the method of Lemma B.2 again, we can move the 2 smallest disks from the peg they're on on to the ending peg, which leads to the arrangement we wanted. ☐

Lemma B.5. *The Towers of Hanoi puzzle with* 4 *disks has a solution.*

Proof. By Lemma B.4, there is a way to move the 3 smallest disks to a different peg. Using this method, move the 3 smallest disks from the starting peg to a third peg: neither the starting nor the ending peg. (In this process, the largest disk won't get in the way, because any other disks can be placed on top of it.)

Now the largest disk is free, and the ending peg is empty, so the largest disk can be moved to the ending peg.

Finally, using the method of Lemma B.2 again, we can move the 3 smallest disks from the peg they're on on to the ending peg, which leads to the arrangement we wanted. ☐

Question:	When you compare the proofs of Lemma B.3, Lemma B.4, and Lemma B.5, what do you observe?
Answer:	They are almost identical: they only differ in the number used and in the lemma referenced. (In the case of Lemma B.3, some plural nouns become singular, but that's just a quirk of the English language.)

It is clear that if we wanted to write a proof of a lemma for 5 disks, or 6 disks, or 64 disks, we could do it by copying and pasting and then changing the numbers. So as soon as we have a template which we can copy and paste, we should accept that all those lemmas are true: we know how to obtain a proof of any of them.

This is the logic underlying a proof by induction. The template that we would copy and paste is exactly the induction step, with specific numbers replaced by $n-1$ or n.

Question:	In this copying-and-pasting view of induction, why do we need to prove the base case?
Answer:	Because the proof of Lemma B.2 didn't follow the template: at that point, we didn't have a previous lemma we could cite. So the proof by induction must tell us how to prove the first lemma, as well as how to prove every other lemma.

A common modification to the basic induction template is **strong induction**, where we allow each case of the theorem to rely on multiple smaller cases, and not just the immediately preceding case. For example, if the 6-disk solution involved using a 5-disk solution and a 4-disk solution, or if it unpredictably went back to a k-disk solution for some unknown $k \in \{1, 2, 3, 4, 5\}$, then we'd have to use strong induction.

Question:	Why is this still okay?
Answer:	Because there is no circular reasoning. If we take all the cases the n^{th} case depends on, and then all the cases they depend on, and so on, eventually we end up going back to the base case.

I want to devote most of this chapter to discussing the way induction works in graph theory in particular, but first I want to give one last general piece of advice.

A common tip for getting started solving a problem is to try small cases of the problem first. It is easier to solve specific small examples than to solve a general problem, and by looking at the examples, you can try to make guesses about a general solution. What I want to tell you about is a related idea: if you think you might want to use induction to solve a problem, don't look at the small cases on their own! Look at consecutive cases together, trying to get an idea of what your induction step might be by seeing how one small case can be used to get the next.

So, for example, no matter how you want to solve the Towers of Hanoi puzzle, it's reasonable to look at what happens for 2 or 3 disks to start with. If you want to try a proof by induction, you should look at the 2-disk solution and 3-disk solution side by side, and try to draw connections between the two.

In this case, you might find the 2-disk solution inside the 3-disk solution: once, or even twice. You might also arrive at a useful insight by comparing the lengths of the solutions. Hypothetically, the relationship could have been different: the 3-disk solution could be the 2-disk solution with a short sequence added to the end, or inserted in the middle, or with some transformation applied to each step.

B.2 Induction on graphs

In graph theory, induction is often used in a special way that's slightly different from induction in many other areas of math. Instead of proving that a statement is true for all natural numbers n, we often try to prove it for all graphs, or at least for all graphs with some property relevant to the specific problem.

This tends to work out reasonably well for us, because we have at least two good ways of measuring how big a graph is: we can count the vertices, or we can count the edges. The most common technique is to induct on the number of vertices in a graph. Here, we consider "the statement is true for all graphs with n vertices" to be a single case, and advance through these cases by induction on n. (I will say more later about inducting on the number of edges later; it is less common.)

However, because we're grouping together many graphs into one case, we have to be very careful about how we set up the induction. If we're not, we can make mistakes and prove false statements, or give incorrect proofs of true statements. The second possibility is harder to catch, because you won't be able to find any counterexamples. To avoid confusing you too much, I will give you an example of the first possibility: a claim that is definitely false.

Claim B.6 (False claim)**.** *For all $n \geq 3$, if G is an n-vertex graph in which every vertex has degree at least 2, then G must have at least $2n - 3$ edges.*

We know this is false, because the cycle graph C_n is a counterexample. Every vertex of C_n has degree 2, but C_n only has n edges, which is less than $2n - 3$ for large n.

Incorrect proof. We induct on n. When $n = 3$, the claim holds, because we need all 3 edges in a 3-vertex graph to exist in order for the assumption to hold, and $3 = 2(3) - 3$.

Assume the claim holds for all $(n - 1)$-vertex graphs: if every in an $(n - 1)$-vertex graph has degree at least 2, then the number of edges is at least $2(n-1)-3$ or $2n-5$. To get an n-vertex graph, we add a vertex; to give it degree at least 2, we need to add at least two new edges. This results in a graph with at least $(2n - 5) + 2$ or at last $2n - 3$ edges.

By induction, the claim is true for all n. □

Just knowing that this proof is incorrect is not enough. We need to understand why it is incorrect, to make sure that we do not do it again.

Question:	The smallest counterexample to Claim B.6 is C_4. Why does the proof fail to consider C_4?
Answer:	When going from 3-vertex graphs to 4-vertex graphs, we try to add a vertex of degree 2 to a 3-vertex graph with minimum degree 2. But C_4 cannot be obtained in this way: if you remove a vertex from it, you're left with P_3, which is not a graph with minimum degree 2.

To avoid this mistake, I have a general theory to propose, and also a practical solution.

The general theory is this. If the possible cases of our theorem were truly the positive integers $n \geq 3$, they would form a chain:

$$3 \to 4 \to 5 \to 6 \to 7 \to 8 \to 9 \to 10 \to \dots$$

If we start at 3 and follow the arrows, we will eventually get to every possibility.

But in Claim B.6, the set of possible cases is really the set of graphs with minimum degree at least 2. Instead of a chain, we can imagine organizing this set into an infinite diagram, a fragment of which is shown in Figure B.2. Here, the arrows represent possible ways to go from a small graph to a bigger one by adding a new vertex of degree 2.

If this is the induction step we intend to use, then a single 3-vertex base case is not enough: by following arrows from that case, we miss many possibilities. We could conceivably write an induction in this way, but we would need many base cases: all the graphs which are not obtained from a smaller graph by following an arrow. At a bare minimum, this includes the cycle graphs, but eventually there are other examples like this. I don't think this is a reasonable

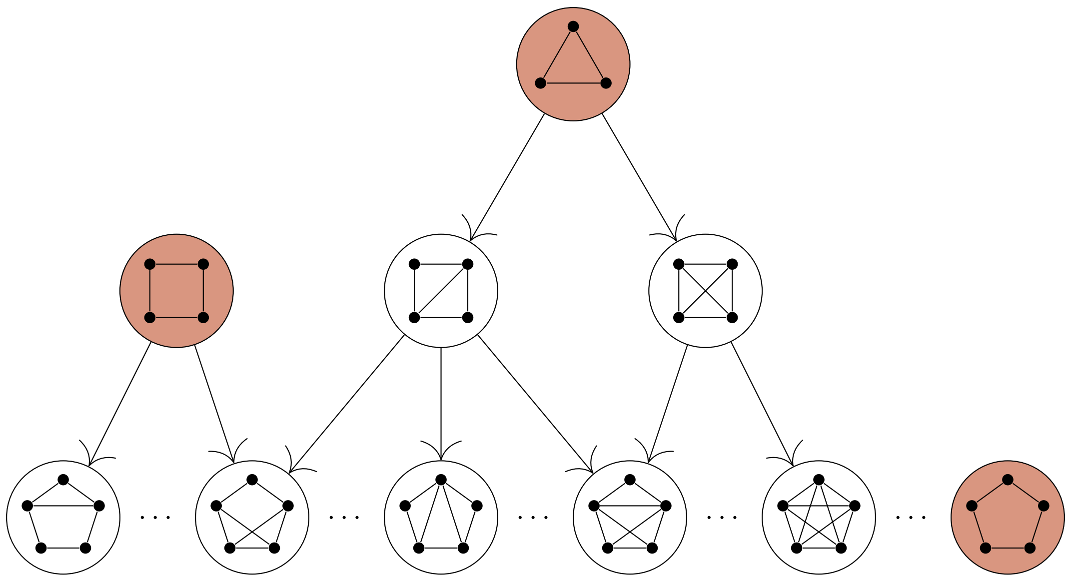


Figure B.2: Organizing the graphs with minimum degree at least 2 according to the induction step of Claim B.6 (not all 5-vertex graphs are included)

way to write an induction proof; I'm just saying that if we wanted to rescue this idea, that's what we'd have to do.

Question:	Is there a 5-vertex graph with minimum degree at least 2, other than C_5, which is not obtained by adding a vertex to a smaller graph of minimum degree at least 2?
Answer:	Yes: we can get another such graph by taking two copies of K_3 that share a vertex of degree 4.

I did, however, promise a practical solution to the problem. This practical solution restricts the way we can write proofs by induction on the vertices of G, allowing only ones that are guaranteed to work. You can think of it as filling in the blanks in the following template:

Theorem B.7 (Induction Template)**.** *If a graph G has property A, it also has property B.*

Proof. We induct on the number of vertices in G. **(Prove a base case here.)**

Now, for some $n > 1$ **(or other lower bound, depending on the base case**), assume that all $(n-1)$-vertex graphs with property A also have property B. Let G be an n-vertex graph with property A. Our goal is to show that G also has property B.

Let x be a vertex of G **(usually chosen by some clever rule you'll have to come up with)**. Then $G - x$ (the graph obtained from G by deleting x and all edges out of x) also has property A **(by an argument related to the way we chose which vertex to delete)**.

By the induction hypothesis, $G - x$ also has property B. When we add back the vertex x, G also has property B **(by another argument you'll have to come up with)**.

By induction, all graphs with property A also have property B. □

Why does this work, when our previous argument didn't? The key is the step "Let G be an n-vertex graph with property A." We didn't make any assumptions about G. Rather, we started from an arbitrary graph with property A; to apply the induction hypothesis, we cooked up a graph $G - x$ which is an $(n-1)$-vertex graph with property A.

Another way to put it is this: using this induction template prevents the induction from having the kind of structure shown in Figure B.2, by forcing each graph G to have a previous case $G - x$ it is obtained from. Only the graphs covered in the base case are exempt from this.

Using the induction template takes practice to master. (A good example of its use early on in this book is in the proof of Lemma 4.6.) However, it is also helpful, because it is a scaffolding for your proofs by induction: it means you don't have to start from scratch.

Earlier, I mentioned induction on the number of edges of a graph. This is only done a few times in this book:

- When proving the handshake lemma (Lemma 4.1) and later when proving its directed version, Lemma 7.1.
- When proving Theorem 8.3 on cycle decompositions.
- When proving Theorem 10.1 on the number of connected components in a graph with m edges.
- When proving Euler's formula (Theorem 21.4) for plane embeddings.

We can use a variant of the induction template here, deleting an edge xy rather than a vertex x. (The proof of Theorem 8.3 uses strong induction: we delete multiple edges at once.) Repeatedly deleting edges leaves a graph with no edges, but the same number of vertices; all such graphs should be our base cases.

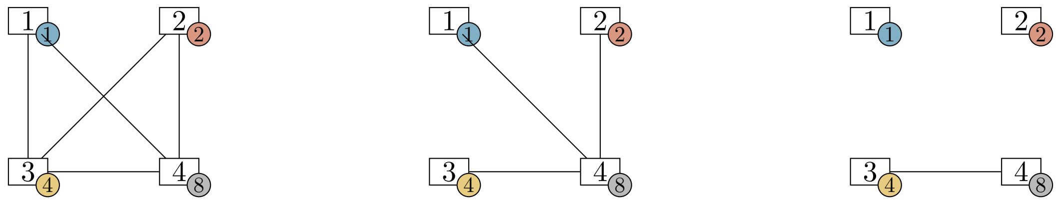
B.3 Inductive definitions

Some structures in graph theory are important because they make it particularly easy to write a proof by induction. There are two that I can point to in this book:

- Trees, which are introduced in Chapter 9. Lemma 10.6 tells us that if T is a tree and x is a degree-1 vertex, then $T - x$ is a tree; moreover, we know from Lemma 10.5 that every tree with multiple vertices has some degree-1 vertices to use Lemma 10.6 with.

  Used in reverse, these two lemmas give us an "inductive definition" of a tree: a tree is either a graph with 1 vertex, or a graph obtained from a smaller tree by adding a new vertex of degree 1.

- 2-connected graphs, which are introduced in Chapter 25. Theorem 25.4 tells us that every 2-connected graph has an ear decomposition, which gives us an inductive definition of a 2-connected graph: it is either a cycle, or a graph obtained from a smaller 2-connected graph by adding an ear.



(a) $t = 4$ (b) $t = 8$ (c) $t = 11$

Figure B.3: 4-vertex threshold graphs with three thresholds t

I don't want to spoil the details of these objects if you haven't read those chapters, even if you're dying to know what it means to add an ear to a graph. But what do I mean by an inductive definition?

Well, in both cases, we are saying something like, "A (type of graph) is either a (base case) or obtained from a smaller (same type of graph) by adding a (small modification)." This is what I mean by an **inductive definition**. You can see how it resembles the structure of a proof by induction—it also makes it easy to write one.

Question:	How can we write a proof by induction using an inductive definition?
Answer:	The base case of the induction is to check that our theorem is true in the base case of the definition.
	The induction step is to check that if the theorem is true for a graph of this type, it remains true after the small modification.

I want to show you how inductive definitions work in graph theory without getting into either of the advanced examples we learn about elsewhere in the book. So I will give you a different example: a family of graphs called *threshold graphs*. These were defined by Václav Chvátal and Peter Hammer in 1977 to study when multiple inequalities with $\{0,1\}$-valued variables can be combined into one [1].

Threshold graphs have two definitions. One is the definition they get their name from. A graph G is a threshold graph if there is a function $f\colon V(G) \to \mathbb{R}$ and a value $t \in \mathbb{R}$ such that two vertices x and y are adjacent if and only if $f(x) + f(y) > t$. In other words, adjacency is defined by the sum of vertex labels exceeding a threshold. Figure B.3 shows three examples of 4-vertex threshold graphs: here, the values of f are $0.3, 0.6, 1.2, 2.4$ in all three diagrams, but we require different thresholds t for an edge to exist.

The other definition is the inductive definition. A graph G is a threshold graph if it has only one vertex, or if it can be obtained from a smaller threshold graph in one of two ways:

- adding a new vertex of degree 0, or
- adding a new vertex adjacent to all existing vertices.

Let's practice induction by proving that we get the same class of graphs by either definition!

After I wrote down the proof below for the first time, I realized that I spent half a page just on writing "threshold graph by the ... definition" many times. So to abbreviate, let's call G a *thres graph* if it satisfies the inequality definition, and a *hold graph* if it satisfies the inductive definition.

Proposition B.8. *A graph G is a thres graph if and only if it is a hold graph.*

Proof. We will need two proofs by induction, one for each direction of the proof.

First, let's show that if graph G is a hold graph, we can define a function $f\colon V(G) \to \mathbb{R}$ to make it a thres graph. For simplicity, we'll use the threshold $t = 0$.

Our base case here is the 1-vertex graph. We can give the single vertex any value of f we like, and it will be a thres graph, because there isn't a pair of vertices for which to check it.

Suppose now that for some hold graph G, we've defined a function f that also makes it a thres graph. We now need to show that with either of the modifications we do, we can extend f.

- Suppose we add a vertex x to G which is not adjacent to any vertex in $V(G)$, getting a bigger hold graph G'.

  Then to extend f to a function $V(G') \to \mathbb{R}$, we define

  $$f(x) := \min\{-1 - f(y) : y \in V(G)\}.$$

  For each $y \in V(G)$, any value of $f(x)$ equal to or less than $-1 - f(y)$ is enough to make sure that $f(x) + f(y) \le -1$, preventing an edge xy; we choose $f(x)$ to be the smallest of these values.

- Suppose we add a vertex x to G which is adjacent to every vertex in $V(G)$, getting a bigger hold graph G''.

  Then to extend f to a function $V(G'') \to \mathbb{R}$, we define

  $$f(x) := \max\{1 - f(y) : y \in V(G)\}.$$

  For each $y \in V(G)$, any value of $f(x)$ at least $1 - f(y)$ is enough to make sure that $f(x) + f(y) \ge 1$, ensuring an edge xy: we choose $f(x)$ to be the largest of these values.

In both cases, the larger hold graph continues to be a thres graph with the extended f. We conclude that as we build bigger and bigger hold graphs from the 1-vertex graph, they always remain thres graphs, and therefore by induction, all hold graphs are thres graphs.

That's one part done. Now we show that if G is a thres graph, it is also a hold graph. Since we're trying to establish the inductive definition, we won't be able to use it as a model for induction; instead, we'll use the induction template.

We induct on the number of vertices in G. When G is a 1-vertex thres graph, it is also a hold graph by the base case of the inductive definition.

Now, for some $n > 1$, assume that all $(n-1)$-vertex thres graphs are also hold graphs. Let G be a thres graph, defined using a function $f\colon V(G) \to \mathbb{R}$ and a threshold t.

Question:	For which vertices x is $G - x$ a thres graph?
Answer:	All of them: we can just keep the same function f (with fewer inputs) and the same threshold t.

So we won't be deleting a vertex carefully to make sure we still have a thres graph; we'll be deleting a vertex carefully to make it's easy to check the definition of a hold graph when we put it back.

I propose two options for the deletion. Let x be the vertex of G such that $f(x)$ is as small as possible, and let y be the vertex of G such that $f(y)$ is as large as possible.

Question:	If $f(x) + f(y) \leq t$, what do we know about x?
Answer:	Since not even $f(y)$ is big enough for the sum with $f(x)$ to exceed the threshold, we know $f(x) + f(z) \leq t$ for all vertices z, so x has no neighbors in G.

In this case, consider the thres graph $G - x$. By the induction hypothesis, $G - x$ is also a hold graph. But G is obtained from $G - x$ by adding a new vertex x of degree 0, which means G is also a hold graph.

Question:	If $f(x) + f(y) > t$, what can we do instead?
Answer:	Since even $f(x) + f(y)$ is bigger than t, and $f(x)$ is the smallest value of f there is, we know $f(z) + f(y) > t$ for all vertices z. Therefore y is adjacent to all other vertices of G.

In this case, consider the thres graph $G - y$. By the induction hypothesis, $G - y$ is also a hold graph. But G is obtained from $G - y$ by adding a new vertex y adjacent to all existing vertices, which means G is also a hold graph.

In both cases, we prove G is a hold graph, completing the induction step. By induction, all thres graphs are hold graphs. □

We can compare the incorrect proof of the false Claim B.6 to the way that we used an inductive definition in this proof. Actually, both proofs are written in the same style: we start with a base case and then we grow it. The difference is that in the case of threshold graphs, the inductive definition guarantees that every other threshold graph can be obtained from a 1-vertex graph. (For a demonstration of this, see Figure B.4.) In other words, one base case is all we need!

B.4 Recursive constructions

Let's say that a sequence of graphs $G_1, G_2, G_3, \dots$ has a **recursive construction** if we have a fixed procedure by which we can build G_n out of G_{n-1} for all $n > 1$. For example, the sequence $Q_1, Q_2, Q_3, \dots$ of hypercube graphs defined in Chapter 4 has a recursive construction. For all n, we can build Q_n from Q_{n-1} by the following procedure:

1. Start with two disjoint copies of Q_{n-1}.

   If we think of the vertices of Q_{n-1} as $(n-1)$-bit strings, then we can say that the first copy of Q_{n-1} has a vertex $x0$ (x, followed by a 0) for every vertex $x \in V(Q_{n-1})$, and the second copy of Q_{n-1} has a vertex $x1$ for every vertex $x \in V(Q_{n-1})$.

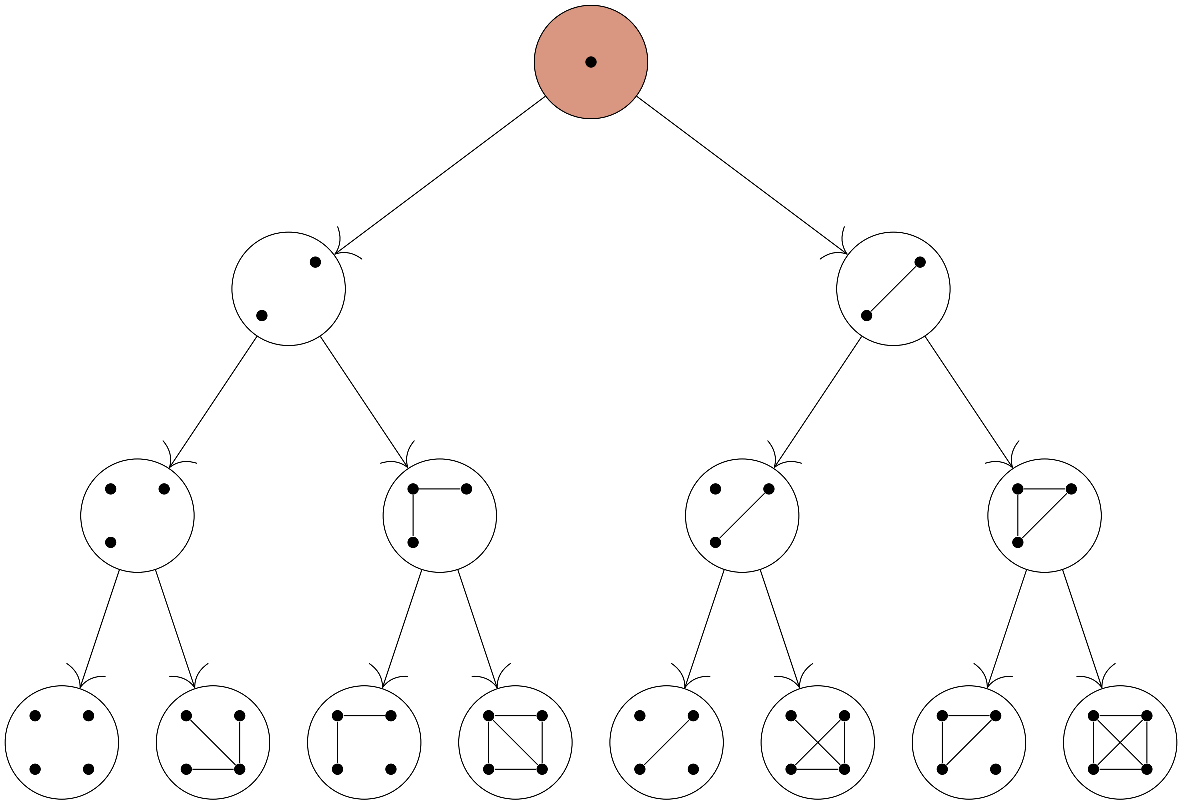


Figure B.4: Building all threshold graphs from a 1-vertex graph

2. In addition to the edges that already exist within each copy of Q_{n-1}, add an edge $\{x0, x1\}$ for all $x \in V(Q_{n-1})$: this is an edge between the two vertices corresponding to x in the two copies of Q_{n-1}.

This is easier to see in a diagram. Figure B.5 shows how we build Q_4 using this recursive construction: the two copies of Q_3 are shown on the left (with a 0 at the end of every vertex) and on the right (with a 1 at the end of every vertex), and the dashed edges are the edges added between the copies.

Other notable examples of recursive constructions in this book include:

- The Mycielski graphs defined in Chapter 19: each graph is the Mycielskian of the previous graph.
- The de Bruijn digraphs defined in Chapter 20: by Proposition 20.3, for all $k \geq 1$, each digraph in the sequence $B(k, 1), B(k, 2), B(k, 3), \ldots$ is the line digraph of the previous graph.
- The iterated barycentric subdivisions used as an example in Chapter 21: to go from each plane embedding to the next, take the barycentric subdivision of every triangular face.

  In general, the word "iterated" in a mathematical term often indicates a recursive construction.

Why do I mention recursive constructions here? You can probably guess why: induction is a good way to prove properties of a sequence of graphs with a recursive construction.

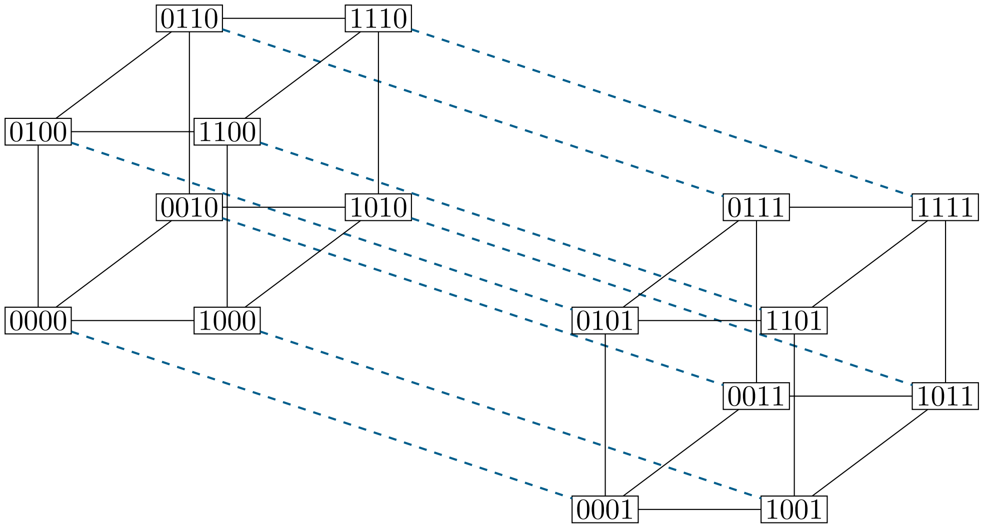


Figure B.5: Using a recursive construction to build Q_4 from two copies of Q_3

Let's start with a simple example, where the proof by induction is actually much more complicated than what we need.

Proposition B.9. *For all $n \geq 1$, the hypercube graph Q_n has 2^n vertices and $n \cdot 2^{n-1}$ edges.*

Proof. We induct on n. When $n = 1$, the hypercube graph Q_1 has two vertices (0 and 1) and one edge between them. Since $2^1 = 1$ and $1 \cdot 2^{1-1} = 1$, both formulas we want to prove are true for $n = 1$.

Now assume, for some $n > 1$, that Q_{n-1} has 2^{n-1} vertices and $(n-1) \cdot 2^{n-2}$ edges. When we construct Q_n recursively, we take two disjoint copies of Q_{n-1}, so we get $2 \cdot 2^{n-1}$ or 2^n vertices. There are $(n-1) \cdot 2^{n-2}$ edges in each copy of Q_{n-1}, and 2^{n-1} edges added between them (one for each vertex of Q_{n-1}), for a total of

$$(n-1) \cdot 2^{n-2} + (n-1) \cdot 2^{n-2} + 2^{n-1} = n \cdot 2^{n-1}$$

edges.[3]

We confirm that if Q_{n-1} has the number of vertices and edges we wanted, then so does Q_n. By induction, the formulas in the proposition hold for all $n \geq 1$. □

Question:	What's the much simpler way to prove that Q_n has $n \cdot 2^{n-1}$ edges?
Answer:	Use the handshake lemma! There are 2^n vertices, and each vertex has degree n, so the sum of degrees is $n \cdot 2^n$, and to get the number of edges, we divide by 2.

[3]It can be tempting to slack off when verifying this identity, since we know what we should get when we simplify if the proof by induction is going to work. Try to resist this temptation and actually do the algebra, or else you risk proving something false one day.

Question:	If we didn't know the formula $n \cdot 2^{n-1}$ ahead of time, could we still use a proof by induction?
Answer:	Yes, but we'd have to solve a recurrence relation to figure out the formula, first. If $f(n)$ is the number of edges, then the argument in our induction step tells us that $f(n) = 2 \cdot f(n-1) + 2^{n-1}$, with $f(1) = 1$.

Here's a quick algebraic way to solve the recurrence relation: first, divide through by 2^n to get $\frac{f(n)}{2^n} = \frac{f(n-1)}{2^{n-1}} + \frac{1}{2}$. Now, each occurrence of $f(k)$ for any k shows up divided by 2^k, so we can define $g(n) = \frac{f(n)}{2^n}$. This has the initial value $g(1) = \frac{f(1)}{2} = \frac{1}{2}$ and a much simpler recurrence relation: $g(n) = g(n-1) + \frac{1}{2}$. If we start with $\frac{1}{2}$ and add $\frac{1}{2}$ each time, then we should get $g(n) = \frac{n}{2}$, and so $f(n) = g(n) \cdot 2^n = n \cdot 2^{n-1}$.

This trick can be used to solve many recurrence relations, but first you have to figure out the right thing to multiply or divide by in order to simplify the recurrence relation.

Here is a second example of induction on Q_n, for which we'll have to work harder.

Proposition B.10. *For all $n \geq 1$, the diameter of Q_n (the largest distance between any two of its vertices) is n.*

Proof. When $n = 1$, Q_n is still the graph with two vertices and 1 edge; this has diameter 1, because the two vertices are at distance 1 from each other.

Now assume, for some $n > 1$, that Q_{n-1} has diameter $n-1$; to prove the induction step, we would like to prove that Q_n has diameter n.

Question:	On the level of walks in a graph, what does our induction hypothesis tell us, and what do we need to prove?
Answer:	We know that for every pair x, y of vertices in Q_{n-1}, there is an $x - y$ walk of length at most $n-1$, and that there is some pair x, y for which no shorter walk exists. We want to show that for every pair x, y of vertices in Q_n, there is an $x - y$ walk of length at most n, and we want to find some pair x, y for which we prove that no shorter walk exists.

First, let's prove that walks of length at most n exist between every pair of vertices in Q_n. If we take two vertices in the same copy of Q_{n-1}, then there is a walk of length at most $n-1$ between them, by the induction hypothesis. So suppose we take two vertices in different copies: without loss of generality, vertices of the form $x0$ and $y1$.

By the induction hypothesis, there is an $x-y$ walk in Q_{n-1} of length at most $n-1$; equivalently, an $x0 - y0$ walk in the first copy of Q_{n-1} inside Q_n. By taking that $x0 - y0$ walk, and adding the vertex $y1$ to the end (which is adjacent to $y0$), we get an $x0 - y1$ walk whose length is only increased by 1: the length is most n, as we wanted.

To find a pair of vertices at distance n, we begin with other half of our induction hypothesis: that Q_{n-1} contains two vertices x, y for which all $x - y$ walks have length at least $n-1$.

Question:	Using x and y, which vertices in Q_n should we pick that we expect to be at distance n?
Answer:	Vertices $x0$ and $y1$, for example: these are as far away as possible and also in different copies of Q_{n-1}.

We can think of a walk in Q_n is a sequence of steps where at each step, we change some coordinate. To get from $x0$ to $y1$, at least once, we need to change the last coordinate, going from $z0$ to $z1$ for some z. Let's just skip all steps where we do that: we will now get a walk from $x0$ to $y0$ in the first copy of Q_{n-1}, and it will be shorter by at least one step.

Question:	What is the motivation for doing this?
Answer:	Our induction hypothesis is an assumption about all $x - y$ walks, so in order to use it, we first need to obtain an $x - y$ walk to apply it to!

This new $x0 - y0$ walk corresponds to an $x - y$ walk in Q_{n-1}, so it has length at least $n - 1$, by our induction hypothesis. The $x0 - y1$ walk has at least one step we skipped, so it has length at least n, completing our lengthy induction step.

We've proved that if Q_{n-1} has diameter $n - 1$, then Q_n has diameter n; by induction, Q_n has diameter n for all $n \geq 1$. □

B.5 Practice problems

1. Let's take a closer look at Theorem B.1.

   a) For $n = 3$ disks, write down the steps of the solution that the proof gives us.

   b) In general, in terms of n, how many steps are there in the solution?

   c) It turns out that the number in part (b) is the least number of steps required to move all the disks from one peg to another. Prove this claim by induction.

2. The sequence $F_0, F_1, F_2, \ldots$ of Fibonacci numbers is defined by $F_0 = 0$, $F_1 = 1$, and the recurrence relation $F_n = F_{n-1} + F_{n-2}$ for all $n \geq 2$.

   A *matching* is defined in Chapter 13 as a spanning subgraph M where every vertex has degree 0 or 1; equivalently, the edges $E(M)$ share no endpoints.

   a) Prove that for all $n \geq 1$, the path graph P_n has F_{n+1} matchings.

   b) Prove that for all $n \geq 3$, the cycle graph C_n has $F_{n+1} + F_{n-1}$ matchings.

3. Prove by induction on n that for all $n \geq 5$, there exists a graph with n vertices and $2n - 4$ edges which has minimum degree 2 and maximum degree 4.

4. Prove that the complement of every threshold graph is also a threshold graph.

5. Let's say that a graph G has "comparable neighborhoods" if, for all $x, y \in V(G)$, either vertex x is adjacent to all the neighbors of y, or vertex y is adjacent to all the neighbors of x. (That is, the *neighborhoods* $N(\{x\})$ and $N(\{y\})$ are *comparable sets*, following definitions in Chapter 15.)

   a) Prove that every threshold graph has comparable neighborhoods.

   b) In the bottom level of Figure B.4, 8 of the 11 possible four-vertex graphs are shown (up to isomorphism). The missing 3 graphs are exactly the 3 graphs that do not have comparable neighborhoods. What are they?

   c) Let G be a graph with comparable neighborhoods. Prove that it either has a vertex adjacent to every other vertex, or a vertex with no neighbors.

   d) Prove by induction that if G has comparable neighborhoods, it is a threshold graph.

6. Let a "quadsum graph" be a graph on at least 4 vertices which is either a 4-vertex cycle or a union $G \cup H$ where G and H are quadsum graphs with 2 vertices and 0 edges in common.

   Prove that all n-vertex quadsum graphs have the same number of edges, and find that number as a function of n.

7. Let a "Hanoi integer" be an integer in which no two adjacent digits are equal (such as 123456, or 271828, but not 144702). We define two operations on Hanoi integers:

   A "tweak" changes the last digit to anything else that's not the same as the previous digit: for example, we may change 123456 to 123450.

   A "twiddle" takes the longest suffix of two digits alternating, as $\ldots xyxy$, and switches the two digits: we may change 123456 to 123465, or 271828 to 271282, or 363636 to 636363. Twiddles are forbidden if they would put a 0 at the beginning of the integer.

   a) Prove that you can change any n-digit Hanoi integer to any other with a combination of $2^n - 1$ tweaks and twiddles.

   b) Prove that $2^n - 1$ operations are required to get from a number written only with the digits 0–4 to a number written only with the digits 5–9.

C License

d. Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

e. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

f. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

g. License Elements means the license attributes listed in the name of a Creative Commons Public License. The License Elements of this Public License are Attribution and Share-Alike.

h. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

i. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

j. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

k. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

l. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

m. You means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

Section 2 – Scope.

a. **License grant**.

   1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

      A. reproduce and Share the Licensed Material, in whole or in part; and

B. produce, reproduce, and Share Adapted Material.

2. **Exceptions and Limitations**. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. **Term**. The term of this Public License is specified in Section 6(a).

4. **Media and formats; technical modifications allowed**. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. Downstream recipients.

   A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

   B. Additional offer from the Licensor – Adapted Material. Every recipient of Adapted Material from You automatically receives an offer from the Licensor to exercise the Licensed Rights in the Adapted Material under the conditions of the Adapter's License You apply.

   C. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. **Other rights**.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. **Attribution**.

1. If You Share the Licensed Material (including in modified form), You must:
   A. retain the following if it is supplied by the Licensor with the Licensed Material:
      i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
      ii. a copyright notice;
      iii. a notice that refers to this Public License;
      iv. a notice that refers to the disclaimer of warranties;
      v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
   B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
   C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.
2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

b. **ShareAlike**.

In addition to the conditions in Section 3(a), if You Share Adapted Material You produce, the following conditions also apply.

1. The Adapter's License You apply must be a Creative Commons license with the same License Elements, this version or later, or a BY-SA Compatible License.
2. You must include the text of, or the URI or hyperlink to, the Adapter's License You apply. You may satisfy this condition in any reasonable manner based on the medium, means, and context in which You Share Adapted Material.
3. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, Adapted Material that restrict exercise of the rights granted under the Adapter's License You apply.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material, including for purposes of Section 3(b); and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

a. Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.

b. To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

   1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

   2. upon express reinstatement by the Licensor.

   For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Bibliography

[1] Václav Chvátal and Peter Hammer. "Aggregation of inequalities in integer programming". In: *Annals of Discrete Mathematics* 1 (1977), pp. 145–162.

[2] Richard Hammack. *Book of Proof*. 2018. ISBN: 978-0-9894721-2-8. URL: https://richardhammack.github.io/BookOfProof/.

[3] Clive Newstead. *An Infinite Descent into Pure Mathematics*. 2024. ISBN: 978-1-950215-00-3. URL: https://infinitedescent.xyz/.

[4] George Pólya. *How to Solve It*. Princeton University Press, 1945. ISBN: 9780691164076.

[5] Princeton University Math Club. *2019 Princeton University Mathematics Competition*. 2019. URL: https://pumac.princeton.edu/archives/#2019 (visited on 11/13/2025).