This problem set – the last one purely on discrete mathematics – is designed as a cumulative review of the topics we’ve covered so far and a proving ground to try out your newfound skills with mathematical induction. The problems here span all sorts of topics – higher dimensions, tiling problems, and games – and we hope that it serves as a fitting coda to our whirlwind tour of discrete math!

We recommend that you read Handout #28, “Guide to Induction,” before starting this problem set. It contains a lot of useful advice about how to approach problems inductively, how to structure inductive proofs, and how to not fall into common inductive traps. Additionally, before submitting, be sure to read over Handout #29, the “Induction Proofwriting Checklist,” for a list of specific things to watch for in your solutions before submitting.

As a note on this problem set – normally, you’re welcome to use any proof technique you’d like to prove results in this course. On this problem set, we’ve specifically asked on some problems that you prove a result inductively. For those problems, you should prove those results using induction or complete induction, even if there is another way to prove the result. (If you’d like to use induction in conjunction with other techniques like proof by contradiction or proof by contrapositive, that’s perfectly fine.)

As always, please feel free to drop by office hours, visit Piazza, or send us emails if you have any questions. We’d be happy to help out.

Good luck, and have fun!

Due Friday, November 1st at 2:30PM.
Problem One: Recurrence Relations

A **recurrence relation** is a way of defining an infinitely long sequence of numbers. A recurrence relation specifies the value of the first term or terms of the sequence, then defines the remaining entries from the previous terms. For example, here’s a simple recurrence relation:

\[ a_0 = 1 \quad a_{n+1} = 2a_n \]

The first terms of this sequence are given as follows:

- \( a_0 = 1 \), since that’s what the first rule says.
- \( a_1 = 2 \), since the second rule says that \( a_1 = 2a_0 = 2 \cdot 1 = 2 \).
- \( a_2 = 4 \), since the second rule says that \( a_2 = 2a_1 = 2 \cdot 2 = 4 \).
- \( a_3 = 8 \), since the second rule says that \( a_3 = 2a_2 = 2 \cdot 4 = 8 \).

Extending further, this sequence starts off with the numbers

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, …,

which all happen to be powers of two. It turns out that this isn’t a coincidence – this recurrence relation perfectly describes the powers of two.

i. Prove by induction that for any \( n \in \mathbb{N} \), we have \( a_n = 2^n \).

In case you’re wondering what you’re asked to prove here, you can think of this recurrence relation as a mathematical way of writing out this recursive function:

```c
int a(int n) {
    if (n == 0) return 1;
    return 2 * a(n - 1);
}
```

For any \( n \in \mathbb{N} \), you can compute \( a(n) \) by just running this code, and after doing some computation it will return the value of \( a_n \). What we’re asking you to do is the mathematical equivalent of showing that the value returned by \( a(n) \) is always \( 2^n \). While it might help to think about things in terms of this analogy, your proof should not reference this code and should just use the definitions given in the problem statement.

Perhaps the most famous recurrence relation is the **Fibonacci sequence**, which is defined as follows:

\[ F_0 = 0 \quad F_1 = 1 \quad F_{n+2} = F_n + F_{n+1} \]

The first terms of this sequence are given as follows:

- \( F_0 = 0 \), since that’s what the first rule says.
- \( F_1 = 1 \), since that’s what the second rule says.
- \( F_2 = 1 \), since the third rule says that \( F_2 = F_0 + F_1 = 0 + 1 = 1 \).
- \( F_3 = 2 \), since the third rule says that \( F_3 = F_1 + F_2 = 1 + 1 = 2 \).
- \( F_4 = 3 \), since the third rule says that \( F_4 = F_2 + F_3 = 1 + 2 = 3 \).

The first ten terms of the Fibonacci sequence are 0, 1, 1, 2, 3, 5, 8, 13, 21, 34. (Make sure you see why!)

If you pull out a calculator and compute ratios of consecutive Fibonacci numbers, you’ll find that the ratio tends toward 1.6180339… . This number is the **golden ratio**, denoted \( \phi \) (the Greek letter phi). Its exact value is \( \phi = \frac{1 + \sqrt{5}}{2} \), and \( \phi \) is the positive solution to the quadratic equation \( x^2 = 1 + x \).

There’s a deep connection between Fibonacci numbers and the golden ratio.

ii. Prove, by induction, that \( \phi^{n+1} = \phi \cdot F_{n+1} + F_n \) for all natural numbers \( n \).

While you can solve this problem by substituting \( \phi = \frac{1 + \sqrt{5}}{2} \) and doing a bunch of algebra, you might find it more useful to use the fact that \( \phi \) is a solution to the equation \( x^2 = x + 1 \).

(Continued on the next page)
The golden ratio $\varphi$ has a companion $\bar{\varphi}$ that is the negative root of the quadratic equation $x^2 = 1 + x$. It's given by the exact formula $\bar{\varphi} = \frac{-1 - \sqrt{5}}{2}$, and there's a result similar to the one you proved in part (ii) that says that $\bar{\varphi}^{n+1} = \bar{\varphi} \cdot F_{n+1} + F_n$ for all natural numbers $n$. Feel free to use this result without proving it; the proof is basically the same as the one you proved in part (ii) of this problem.

iii. Prove that $F_{n+1} = \frac{1}{\sqrt{5}} (\varphi^{n+1} - \bar{\varphi}^{n+1})$ for all natural numbers $n$. Note that we did not ask you to prove this using induction.

To clarify what you're being asked to prove here: in part (i) of this problem, you proved that while we could use the recurrence relation for $a_n$ to compute what $a_n$ is for any natural number $n$, we could alternatively compute $a_n$ by computing $2^n$. This problem is similar: we already know one way to compute $F_{n+1}$ using the recurrence relation, and you're proving is that we can instead compute $\frac{1}{\sqrt{5}} (\varphi^{n+1} - \bar{\varphi}^{n+1})$. As a hint, use your result from part (ii).

The result you proved in part (iii) shows that the Fibonacci numbers grow exponentially quickly, which has implications for the design of data structures like AVL trees and Fibonacci heaps, as well as algorithms like Euclid’s algorithm. Want to learn more? Take CS166!

Finding non-recursive definitions for recurrences (often called “solving” recurrences) is useful in the design and analysis of algorithms. To learn more, take CS161, Math 108, or consider reading through the excellent textbook *Concrete Mathematics* by Graham, Knuth, and Patashnik.

**Problem Two: The Circle Game**

You have a circle with $2n$ arbitrarily-chosen points on its circumference for some natural number $n \geq 1$. Of the $2n$ points, $n$ are labeled +1, and the remaining $n$ are labeled -1. One sample circle with eight points, of which four are labeled +1 and four are labeled -1, is shown below.

Here’s a game you can play. Pick one of the $2n$ points as your starting point, then move clockwise around the circle. You lose the game if at any point on you pass through more -1 points than +1 points. You win the game if you get all the way back to your starting point without losing. For example, if you start at point A, the game would go like this:

Start at A: +1.
Pass through B: +2.
Pass through C: +1.
Pass through D: 0.
Pass through E: -1. *(You lose.)*

If you started at point G, the game would go like this:

Start at G: -1 *(You lose.)*

However, if you started at point F, the game would go like this:

Start at F: +1.
Pass through G: 0.
Pass through H: +1.
Pass through A: +2.
Pass through B: +3.
Pass through C: +2.
Pass through D: +1.
Pass through E: +0.
Return to F. *(You win!)*

No matter which $n$ points are labeled +1 and which $n$ points are labeled -1, there is always at least one point you can start at to win the game. Prove, by induction, that this fact is true for any $n \geq 1$.

*Check the Guide to Induction and Inductive Proofwriting Checklist before starting this one.*
Problem Three: It’ll All Even Out

Our very first proof by induction was the proof that for any natural number $n$, we have that

$$2^0 + 2^1 + 2^2 + \ldots + 2^{n-1} = 2^n - 1.$$ 

This result is still true for the case where $n = 0$, since in that case the sum on the left-hand side of the equation is the empty sum of zero numbers, which is by definition equal to zero. It’s also true for the case where $n = 1$; in that case, the sum on the left-hand side of the equality just has a single term in it ($2^0$) and the right-hand side has the same value.

Below is a proof by complete induction of an incorrect statement about what happens when you sum up zero or more real numbers:

**Incorrect! Theorem:** The sum of any number of real numbers is even.

**Incorrect! Proof:** Let $P(n)$ be the statement “the sum of any $n$ real numbers is even.” We will prove by complete induction that $P(n)$ holds for all $n \in \mathbb{N}$, from which the theorem follows.

As a base case, we prove $P(0)$, that the sum of any 0 real numbers is even. The sum of any zero numbers is the empty sum and is by definition equal to 0, which is even. Thus $P(0)$ holds.

For our inductive step, assume for some arbitrary $k \in \mathbb{N}$ that $P(0), \ldots, P(k)$ are true. We will prove that $P(k+1)$ is true, meaning that the sum of any $k+1$ real numbers is even. To do so, let $x_1, x_2, \ldots, x_k$, and $x_{k+1}$ be arbitrary real numbers and consider the sum

$$x_1 + x_2 + \ldots + x_k + x_{k+1}.$$ 

We can group the first $k$ terms and the last term independently to see that

$$x_1 + x_2 + \ldots + x_k + x_{k+1} = (x_1 + x_2 + \ldots + x_k) + (x_{k+1}).$$ 

Now, consider the sum $x_1 + x_2 + \ldots + x_k$ of the first $k$ terms. This is the sum of $k$ real numbers, so by our inductive hypothesis that $P(k)$ is true we know that this sum must be even. Similarly, consider the sum $x_{k+1}$ consisting of just the single term $x_{k+1}$. By our inductive hypothesis that $P(1)$ is true, we know that this sum must be even.

Overall, we have shown that $x_1 + x_2 + \ldots + x_k + x_{k+1}$ can be written as the sum of two even numbers (namely, $x_1 + x_2 + \ldots + x_k$ and $x_{k+1}$), so $x_1 + x_2 + \ldots + x_k + x_{k+1}$ is even. Thus $P(k+1)$ is true, completing the induction. ■

Of course, this result has to be incorrect, since there are many sums of real numbers that don’t evaluate to an even number. The question, then, is where the proof breaks down.

i. The proof defines a predicate $P(n)$, then uses complete induction to prove $P(n)$ holds for all $n \in \mathbb{N}$. Is $P(n)$ actually a predicate? Does it pass the Induction Proofwriting Checklist? Is it actually the case that, if $P(n)$ is true for all $n \in \mathbb{N}$, then the theorem in question is true? If any of your answers are “no,” explain why, pointing out, specifically, what the proof does wrong.

ii. Is $P(0)$ true? Is the base case of this proof written correctly? If not, point out a specific claim it makes that’s incorrect and explain why it’s incorrect.

*We aren’t looking for “sins of omission” here in the sense of “the proof should have also done X in addition to what it already did.” Rather, we’re looking for “sins of commission” in sense of “the proof does X, and X is incorrect.”*

iii. Is $P(1)$ true? Is the inductive step of this proof written correctly? If not, point out a specific claim it makes that’s incorrect and explain why it’s incorrect.
**Problem Four: Nim**

*Nim* is a family of games played by two players. The game begins with several piles of stones, each of which has zero or more stones in it, shared between the two players. Players alternate taking turns removing any nonzero number of stones from any single pile of their choice. If at the start of a player’s turn all the piles are empty, then that player loses the game.

Prove, by induction, that if the game is played with just two piles of stones, each of which begins with exactly the same number of stones, then the second player can always win the game if she plays correctly.

*Play this game with a partner until you can find a winning strategy. Once you spot the pattern, see if you can find a way to formalize it using induction. Be wary of writing statements of the form “and so on” or “by repeating this;” induction is the proper way to formalize those sorts of ideas.*

*Something to think about – you know that the number of stones in each pile will be decreasing. Can you say how much that number will decrease by? Based on that, what style of proof should you use here?*

**Problem Five: Tiling with Triominoes**

A *right triomino* is an L-shaped tile that looks like this:

Suppose you’re given a $2^n \times 2^n$ grid of squares and want to tile it with right triominoes by covering the grid with triominoes such that

- all triominoes are contained purely within the grid and don’t hang off the sides,
- every square in the grid is completely covered by triominoes, and
- no triominoes overlap.

It’s, unfortunately, never possible to perfectly tile such a board, but, amazingly, it turns out that it is always possible to tile any $2^n \times 2^n$ grid that’s missing exactly one square. It doesn’t matter what $n$ is or which square is removed; there is always a solution to the problem. To the right is a diagram showing how to do this for all $4 \times 4$ grids.

Prove by induction that for any natural number $n$, any $2^n \times 2^n$ grid with any one square removed can be tiled by right triominoes.

*As a note, the fact that a $2^n \times 2^n$ grid missing a square has $4^n - 1$ total squares is true but mostly irrelevant here. A grid of dimension $(4^n - 1) \times 1$ also has $4^n - 1$ squares in it, but that grid, in general, can’t be tiled by right triominoes because they’re only one square wide. In other words, you can’t prove this result simply by counting squares in the grid; the arrangement of those squares matters!*  

*Before you write this proof, try seeing if you can find a nice recursive pattern you can follow that will let you fully tile any such board. You should be able to easily tile any $8 \times 8$ chessboard missing a square with right triominoes before you attempt to write up your answer. Once you can do this, formalize your idea in your answer.*

*Also, is this an “induct up” problem, or an “induct down” problem?*
Problem Six: Dedekind-Infinite Sets

Suppose you have a function $f : A \to A$ from some set $A$ to itself that is injective but \textbf{not} surjective. Knowing nothing more than this, you can conclude that $A$ has to be infinite. This question explores why.

i. Since $f : A \to A$ is not a surjection, the following first-order logic statement about $f$ is \textbf{not} true:

$$\forall x \in A. \exists y \in A. f(y) = x$$

Since the above formula is \textbf{not} true, its negation must be true. Negate the above first-order logic statement and simplify it as much as possible.

The negation that you came up with in part (i) of this problem tells us that there’s an element $x \in A$ with certain properties. We can use this element $x$ to define the following recurrence relation:

$$e_0 = x$$, where $x$ is the element of $A$ singled out above.

$$e_{n+1} = f(e_n)$$

This recurrence relation defines a series of elements $e_0, e_1, e_2, e_3, \ldots$ that continues outward to infinity. Amazingly, each element in the sequence is different from the rest. The rest of this question explores why.

ii. In one sentence, explain why $e_1 \neq e_0$. Then, in one sentence, explain why $e_2 \neq e_0$. Finally, in one sentence, explain why $e_2 \neq e_1$.

To prove that no two terms in this sequence are equal, we’re going to ask you to prove this theorem:

\textbf{Theorem:} For any natural numbers $m$ and $n$ where $m < n$, we have $e_m \neq e_n$.

If this theorem is true, it means, for example, that $e_{137} \neq e_{42}$, since we can plug in $m = 42$ and $n = 137$. Similarly, we know that $e_{103} \neq e_{166}$, which would follow from plugging in $m = 103$ and $n = 166$.

iii. Prove, by induction on $n$, that this predicate $P(n)$ is true for all $n \in \mathbb{N}$:

$$P(n)$$ is the statement “for any $m \in \mathbb{N}$ where $m < n$, we have $e_n \neq e_m$.”

Try generalizing your answers to part (ii) of this problem.

\textit{There are a lot of variables to keep track of, so be careful to scope and introduce them properly. The inductive step of this problem, in particular, would be a great place to write out two columns, one of the things you're assuming, and one of the things that you're proving. The most common class of mistake we tend to see on this problem is mixing up arbitrarily-chosen values with placeholders. One specific thing to keep an eye on: make specific claims about specific variables, and, specifically, be very careful to make sure you aren't using placeholder variables.}

\textit{Is this a problem where you'll induct up? Or induct down?}

You’ve just shown that there must be infinitely many elements in $A$, since the sequence $e_0, e_1, e_2, \ldots$ stretches on forever. And all that follows just from the fact that $f$ is injective but not surjective!

A set $A$ with a function $f : A \to A$ that is injective but not surjective is called \textbf{Dedekind-infinite}. In the early days of set theory, the question arose of how to define what “infinite” meant without referring to natural numbers, and Richard Dedekind proposed this definition, hence the name. Later, it was discovered that this question was far more nuanced than anyone had expected. Want to hear more? Take Math 161!
Problem Seven: A Visit to the Fourth Dimension, Part II

What would it be like to hold a tesseract in your hand? From experience, if you pick up a cube, you can twist it around and it has all sorts of symmetries. From a graph theory perspective, that would lead us to think that $Q_3$ should have many automorphisms, since each automorphism corresponds to a symmetry. So here’s a question: does $Q_4$ have the same sort of symmetries that you’d expect of a cube or a square?

As a refresher from Problem Set Four, the hypercube graph of order $k$, denoted $Q_k$, is defined as follows:

- The nodes of $Q_k$ are the elements of $\wp(\lbrack k \rbrack)$. (Refer to Problem Set Three for the definition of $\lbrack k \rbrack$.)
- There is an edge between a pair of nodes $S$ and $T$ if (and only if) $|S \Delta T| = 1$.

Now, a new definition. A graph $G = (V, E)$ is node-symmetric if, for any $u, v \in V$, there is an automorphism $\sigma$ of $G$ where $\sigma(u) = v$. Intuitively, this means all nodes in $G$ “look the same,” since for any pair of nodes there’s a symmetry of the graph (an automorphism) that makes the first node look like the second.

Prove that for every natural number $k$, the graph $Q_k$ is node-symmetric. You can assume that symmetric difference is commutative ($(S \Delta T) = (T \Delta S)$) and associative ($(S \Delta T) \Delta R = S \Delta (T \Delta R)$).

This problem is much, much easier to solve if you take the time to work through some examples. Can you find an automorphism of $Q_2$ that maps $\emptyset$ to $\{0\}$? To answer that question, try connecting it back to an actual square and its symmetries, see what happens to the corners, and see if you can use that to define an automorphism. There are two ways to do this, one of which can be expressed in terms of a very simple rule (e.g. $\sigma(S) =$ __), while the other is a bit trickier to express. Then, find a different automorphism that maps $\emptyset$ to $\{0, 1\}$, or perhaps an automorphism mapping $\{0\}$ to $\{0, 1\}$. In each case, there are two possible automorphisms that you can choose, one of which will be kinda messy to write out, and one of which will have a very simple rule. Your goal here is to see if you can find a general pattern that maps one node $U$ to another node $V$.

Once you think you have something that works for $Q_2$, see if it also works for $Q_3$! Find an automorphism of $Q_3$ that maps $\{0, 1\}$ to $\{0, 2\}$ by thinking about what that means in terms of symmetries of a cube. Again, there are several options, but one of them will (1) be easy to express symbolically and (2) nicely generalize your automorphisms from above. Then find an automorphism mapping $\{0, 2\}$ to $\{0, 1, 2\}$, or one mapping $\{1\}$ to $\emptyset$, etc. Your goal is to find a general pattern.

Ultimately, you should aim to find a way to, given a node $U$ and a node $V$ of $Q_k$, define an automorphism $\sigma$ of $Q_k$ that maps $U$ to $V$. You should be able to define that rule by filling in the following blank without needing any more space than what’s given below.

$$\sigma(S) = \text{___________}$$

You shouldn’t need a piecewise function.

At this point, you’ve got a general pattern for how to find these sorts of automorphisms, and all that’s left to do is to prove that what you have indeed works. So write down a list of everything you need to prove. You’ll need to show both that what you have is an automorphism (think about what you did on Problem Set Four) and that your automorphism has some other property (which one, exactly?). Break it down further – how do you prove each of those properties? Then, go prove all those properties. In doing so, look back to Problem Set One or Problem Set Two. Perhaps there are some nice results from there you could use here?

Oh, and there’s no need to use induction here. You’ve already shown us that you can do that on the rest of this problem set. 😊

Once you’ve finished this problem, take a step back and admire just how far you’ve come in six weeks! That’s an impressive amount of progress, and you should feel really proud of yourself!
Optional Fun Problem: Egyptian Fractions (Extra Credit)

The Fibonacci sequence mentioned in Problem One is named after Leonardo Fibonacci, an eleventh-century Italian mathematician who is credited with introducing Hindu-Arabic numerals (the number system we use today) to Europe in his book Liber Abaci. This book also contained an early description of the Fibonacci sequence, from which the sequence takes its name.

Liber Abaci also described a method of writing out fractions called Egyptian fractions, which has been employed since ancient times; the Rhind Mathematical Papyrus, composed about 3,500 years ago in Thebes, includes several tables of fractions written out this way.

An Egyptian fraction is a sum of distinct fractions whose numerators are all one (these fractions are called unit fractions). For example, here are some sample Egyptian fraction representations:

$$\frac{2}{3} = \frac{1}{2} + \frac{1}{6} \quad \frac{2}{15} = \frac{1}{10} + \frac{1}{30}$$

$$\frac{7}{15} = \frac{1}{3} + \frac{1}{8} + \frac{1}{120} \quad \frac{2}{85} = \frac{1}{51} + \frac{1}{255}$$

Egyptian fractions are useful for divvying up objects fairly. For example, suppose you have two cakes to distribute to fifteen people – that is, everyone should get a $\frac{2}{15}$ fraction of those cakes. Begin by slicing each cake into tenths and giving each person one ($\frac{1}{10}$). Now, take the remaining tenths you haven’t distributed and cut them into thirds, giving thirtieths of the original cake. Each person then takes one of those ($\frac{1}{30}$). Because $\frac{1}{10} + \frac{1}{30} = \frac{2}{15}$, everyone gets their fair share. Pretty cool, isn’t it?

One way of finding an Egyptian fraction representation of a rational number is to use a greedy algorithm that works by finding the largest unit fraction at any point that can be subtracted out from the rational number. For example, to compute the fraction for $\frac{42}{137}$, we would start off by noting that $\frac{1}{4}$ is the largest unit fraction less than $\frac{42}{137}$. We then say that $\frac{42}{137} = \frac{1}{4} + \left(\frac{42}{137} - \frac{1}{4}\right) = \frac{1}{4} + \frac{31}{548}$.

We then repeat this process by finding the largest unit fraction less than $\frac{31}{548}$ and subtracting it out. This number is $\frac{1}{18}$, so we get

$$\frac{42}{137} = \frac{1}{4} + \left(\frac{42}{137} - \frac{1}{4}\right) = \frac{1}{4} + \frac{1}{18} + \left(\frac{31}{548} - \frac{1}{18}\right) = \frac{1}{4} + \frac{1}{18} + \frac{1}{4932}$$

The largest unit fraction we can subtract from $\frac{5}{4932}$ is $\frac{1}{987}$:

$$\frac{42}{137} = \frac{1}{4} + \frac{1}{18} + \left(\frac{5}{4932} - \frac{1}{987}\right) = \frac{1}{4} + \frac{1}{18} + \frac{1}{987} + \frac{1}{1,622,628}$$

And at this point we’re done, because the leftover fraction is itself a unit fraction.

Prove that the greedy algorithm for Egyptian fractions always terminates for any rational number $r$ in the range $0 < r < 1$ and always produces a valid Egyptian fraction. (A rational number is a real number that can be written as $r = \frac{p}{q}$ for some integers $p$ and $q$ where $q \neq 0$.) That is, the sum of the unit fractions should be the original number, there should only be finitely many fractions, and no unit fraction should be repeated. This shows that every rational number in the range $0 < r < 1$ has at least one Egyptian fraction representation.