Problem Set 2

This second problem set is designed to help you master inductive proofs. It begins with some simpler problems intended to help you adjust to induction, then concludes with applications to infinite series, recursive functions, and contests.

As much as you possibly can, please try to work on this problem set individually. That said, if you do work with others, please be sure to cite who you are working with and on what problems. For more details, see the section on the honor code in the course information handout.

In any question that asks for a proof, you **must** provide a rigorous mathematical proof. You cannot draw a picture or argue by intuition. You should, at the very least, state what type of proof you are using, and (if proceeding by contradiction, contrapositive, or induction) state exactly what it is that you are trying to show. If we specify that a proof must be done a certain way, you must use that particular proof technique; otherwise you may prove the result however you wish.

If you are asked to prove something by induction, you may use either weak induction, strong induction, or the well-ordering principle. You should state your base case before you prove it, and should state what the inductive hypothesis is before you prove the inductive step.

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

This problem set has 142 possible points and nine questions (one checkpoint problem, six main questions, one survey question, and one extra credit problem). It is weighted at 7% of your total grade. The earlier questions serve as a warm-up for the later problems, so do be aware that the difficulty of the problems does increase over the course of this problem set.

Good luck, and have fun!

Checkpoint Questions Due Monday, October 8 at 2:15PM

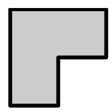
Remaining Questions Due Friday, October 12 at 2:15 PM

Write your solutions to the following problems and submit them by this Monday, October 8th at the start of class. These problems will be graded based on whether or not you submit it, rather than the correctness of your solutions. We will try to get these problems returned to you with feedback on your proof style this Wednesday, October 10th. Submission instructions are at the end of this problem set.

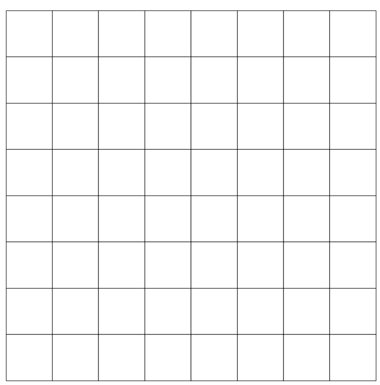
Please make the best effort you can when solving these problems. We want the feedback we give you on your solutions to be as useful as possible, so the more time and effort you put into them, the better we'll be able to comment on your proof style and technique. **Note that this question has three parts.**

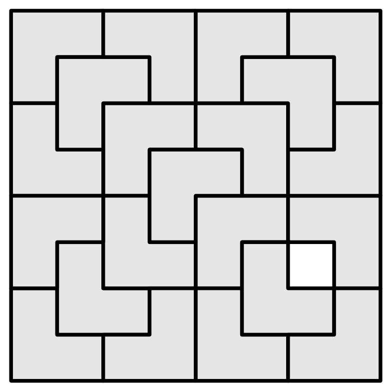
Checkpoint Question: Tiling with Triominoes* (25 Points if Submitted)

Suppose that you are given a set of *right triominoes*, blocks with this shape:



Suppose that you are also given a square grid of size $2^n \times 2^n$ and want to *tile* it with right triominoes by covering the grid with triominoes such that all triominoes are completely on the grid and no triominoes overlap. Here's an attempt to cover an 8×8 grid with triominoes, which fails because not all squares in the grid are covered:





It turns out that it is impossible to completely tile the $2^n \times 2^n$ figure with these triominoes because there are 4^n squares in a $2^n \times 2^n$ figure, but any figure that can be tiled with triominoes must have a multiple of three squares in it. However, 4^n is not a multiple of three. Interestingly, though, if we remove some square from the $2^n \times 2^n$ grid, then there are a total of $4^n - 1$ squares to cover, which *is* a multiple of three.

i. Prove, by induction on n, that $4^n - 1$ is a multiple of three for any $n \in \mathbb{N}$.

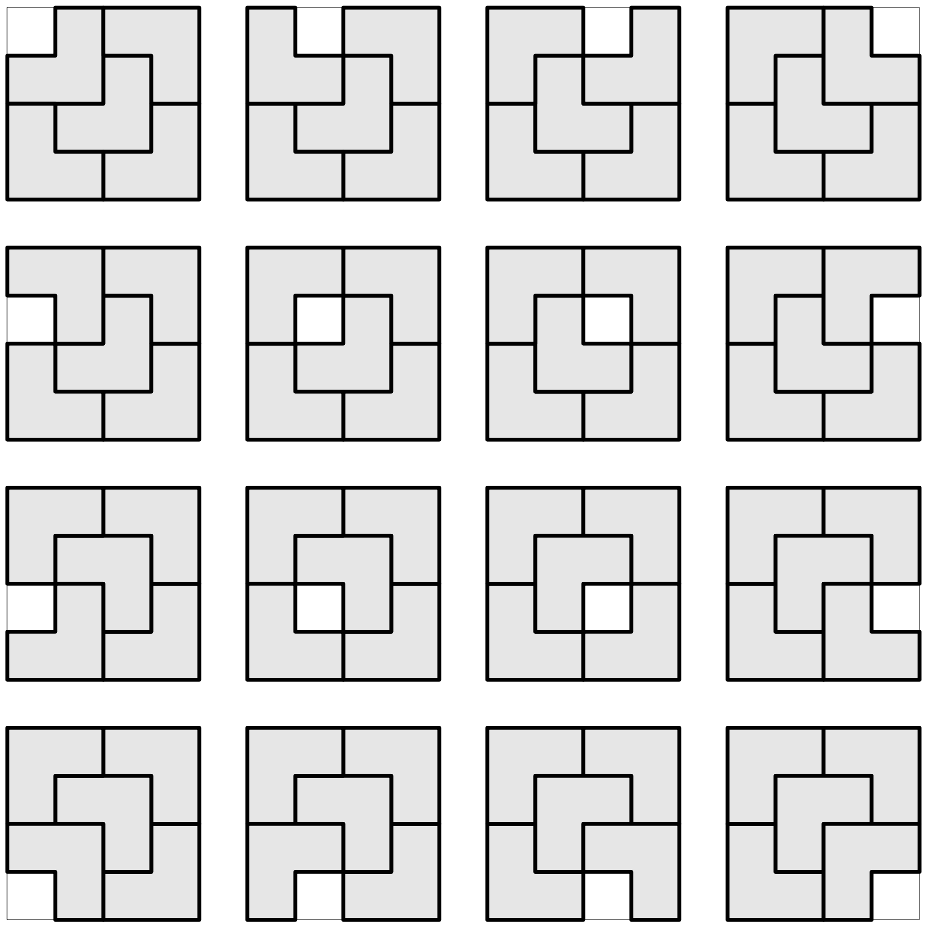
ii. It is *necessary* that a figure have a multiple of three squares in order to be tiled with right triominoes. That is, if a figure does not have a multiple of three squares, then there is no way to tile it with right triominoes. However, it is not true that it is *sufficient* that if a figure has a multiple of three squares in it that it must can be tiled with right triominoes. Give an example of a shape that contains an multiple of three squares, but which cannot be tiled by right triominoes.

(continued on next page)

---

* I originally heard this problem from David Gries of Cornell University. The terminology used here is borrowed from *Discrete Math and its Applications, Sixth Edition* by Kenneth Rosen.

Amazingly, it turns out that it is always possible to tile any $2^n \times 2^n$ grid that's missing exactly one square with right triominoes. It doesn't matter what n is or which square is removed; there is always a solution to the problem. For example, here are all the ways to tile a 4×4 grid that has a square missing:



iii. Prove that any $2^n \times 2^n$ grid with one square removed can be tiled by right triominoes.

The remainder of these problems should be completed and returned by Friday, October 12 at the start of class.

Problem One: Summation Inequalities (8 Points)

Consider two sequences of n numbers $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_n$ where, for any $i \in \mathbb{N}$ such that $1 \le i \le n$, we have $x_i \ge y_i$. For example, we might have sequences like 1, 1, 3, 5, 9 and 0, 1, 1, 2, 3, where each term of the first sequence is at least as large as the corresponding term of the second sequence.

Prove, by induction on n, that $\sum_{i=1}^{n} x_i \ge \sum_{i=1}^{n} y_i$.

Problem Two: Nim (16 points)

Nim is a family of games played by two players. The game begins with several piles of stones. 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 that if the game is played with two piles of stones, each of which begins with the same number of stones, then the second player can always win the game if she plays correctly.

Problem Three: Contract Rummy (16 points)

Contract rummy is a card game for any number of players (usually between three and five) in which players are dealt a hand of cards and, through several iterations of drawing and discarding cards, need to accumulate *sets* and *sequences*. A set is a collection of three cards of the same value, and a sequence is a collection of four cards of the same suit that are in ascending order. The game proceeds in multiple rounds in which players need to accumulate a different number of sets and sequences. The rounds are:

- Two sets (six cards)
- One set, one sequence (seven cards)
- Two sequences (eight cards)
- Three sets (nine cards)
- Two sets and a sequence (ten cards)
- One set and two sequences (eleven cards)
- Three sequences (twelve cards)

Notice that in each round, the requirements are such that the number of cards required increases by one. It's interesting that it's always possible to do this, since the total number of cards must be made using just combinations of three cards and four cards.

Prove, by induction, that any natural number greater than or equal to six can be written as $3m + 4n$ for some natural numbers m and n.

Problem Four: The Harmonic Series (20 Points)

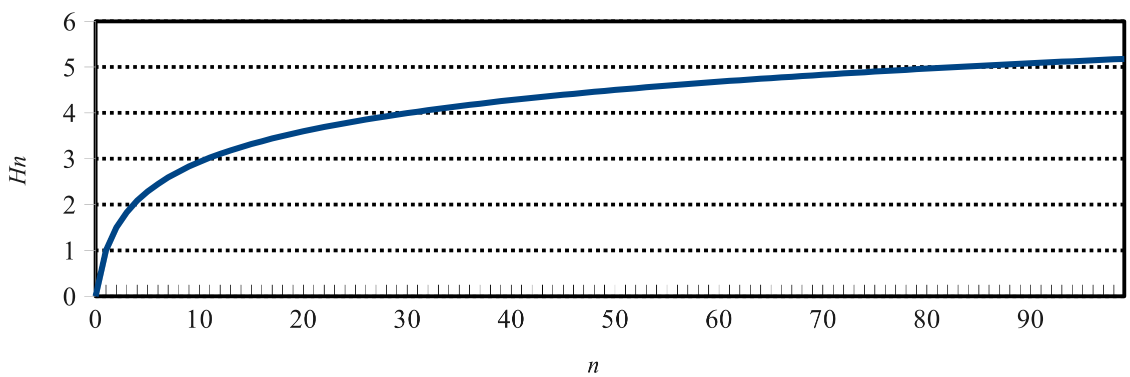
The *harmonic sequence* is the sequence $^1/_1$, $^1/_2$, $^1/_3$, $^1/_4$, … $^1/_n$, …. This sequence has numerous applications in mathematics and computer science. For example, it can be used both to describe the way that strings vibrate in musical instruments and to analyze the expected running time of the quicksort algorithm.

We can sum up the first n terms of the harmonic sequence to get the nth *harmonic number*:

$$H_n = \sum_{k=1}^{n} \frac{1}{k}$$

The first few harmonic numbers are $H_0 = 0$, $H_1 = 1$, $H_2 = {}^3/_2$, $H_3 = {}^{11}/_6$, $H_4 = {}^{25}/_{12}$, …

Since each harmonic number is the sum of progressively more terms of the harmonic series, we know that each harmonic number is bigger than the previous one. However, the terms of the harmonic series only get slightly larger each time; the difference between H_n and H_{n+1} is only $^1/_{n+1}$. As a result, the harmonic numbers grow very slowly, as seen below:



It seems like the harmonic numbers might get closer and closer to some limiting real number without ever crossing it. Amazingly, though, this is not the case. As n goes to infinity, H_n goes to infinity as well. It just does so very slowly.

In this problem, you will prove that the harmonic series diverges (does not converge to some real number) by exploring a proof attributed to fourteenth-century mathematician Nicole Oresme.

i. In the first step of this proof, you will show that the sum of an exponentially large number of terms of the harmonic series becomes appreciably large. Prove, by induction on n, that

$$H_{2^n} \geq \frac{n}{2} + 1$$

That is, the sum of the first 2^n terms of the harmonic series is at least $n / 2 + 1$. This result shows, among other things, that if you were to sum up the first 2^{98} terms of the harmonic series, the result would be at least 50. Please don't try to confirm that by hand. ☺

ii. You can now use your result from part (i) to show that the harmonic series diverges, meaning that there is no number r such that H_0, H_1, H_2, …, converges to r. Specifically, prove the following: There does not exist a real number r such that $H_n \leq r$ for all $n \in \mathbb{N}$.

Problem Five: Repeated Squaring (24 points)

In many applications in computer science, especially cryptography, it is important to compute exponents efficiently. For example, the RSA public-key encryption system, widely used in secure communication, relies on computing huge powers of large numbers. Fortunately, there is a fast algorithm called *repeated squaring* for computing x^y in the special case where y is a natural number.

The repeated squaring algorithm is based on the following function *RS*:

$$RS(x,y)=\begin{cases} 1 & \text{if } y=0 \\ RS(x,y/2)^2 & \text{if } y \text{ is even and } y>0 \\ x\cdot RS(x,(y-1)/2)^2 & \text{if } y \text{ is odd and } y>0 \end{cases}$$

For example, we could compute 2^{10} using $RS(2, 10)$ follows:

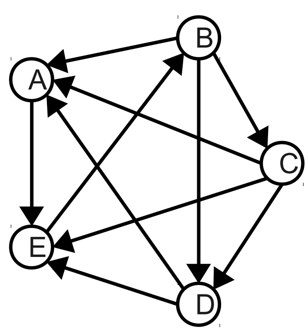
In order to compute $RS(2, 10)$, we need to compute $RS(2, 5)^2$.
In order to compute $RS(2, 5)$, we need to compute $2 \cdot RS(2, 2)^2$.
In order to compute $RS(2, 2)$, we need to compute $RS(2, 1)^2$.
In order to compute $RS(2, 1)$, we need to compute $2 \cdot RS(2, 0)^2$.
By definition, $RS(2, 0) = 1$
so $RS(2, 1) = 2 \cdot RS(2, 0)^2 = 2 \cdot 1^2 = 2$.
so $RS(2, 2) = RS(2, 1)^2 = 2^2 = 4$.
so $RS(2, 5) = 2 \cdot RS(2, 2)^2 = 2 \cdot 4^2 = 32$.
so $RS(2, 10) = RS(2, 5)^2 = 32^2 = 1024$.

The *RS* function is interesting because it can be computed much faster than simply multiplying x by itself y times. Since *RS* is defined recursively in terms of *RS* with the y term roughly cut in half, *RS* can be evaluated using approximately $\log_2 y$ multiplications. (You don't need to prove this).

Prove that for any $x \in \mathbb{R}$ and any $y \in \mathbb{N}$, that $RS(x, y) = x^y$.

Problem Six: Tournament Graphs (28 points)

A *tournament* is a contest among $n > 0$ players. Each player plays a game against each other player, and either wins or loses the game (let's assume that there are no draws). A *tournament graph* is a graph representing the result of a tournament, where each node corresponds to a player and each edge (u, v) means that player u won her game against player v. For example, here is a simple tournament graph for five players:



A *tournament winner* is a player in a tournament who, for each other player, either won her game against that player, or won a game against a player who in turn won against that player. For example, in the above graph, B, C, and E are tournament winners. However, D is not a tournament winner, because he neither won against player C, nor won against anyone who in turn won against C.

Prove that every tournament graph has a tournament winner.

Problem Seven: Course Feedback (5 Points)

We want this course to be as good as it can be, and we'd really appreciate your feedback on how we're doing. For a free five points, please answer the following questions. We'll give you full credit no matter what you write (as long as you write something!), but we'd appreciate it if you're honest about how we're doing.

i. How hard did you find this problem set? How long did it take you to finish? Does that seem unreasonably difficult or time-consuming for a five-unit class?

ii. Did you attend Monday's problem session? If so, did you find it useful?

iii. Did you read the online course notes? If so, did you find them useful?

iv. How is the pace of this course so far? Too slow? Too fast? Just right?

v. Is there anything in particular we could do better? Is there anything in particular that you think we're doing well?

Submission instructions

There are three ways to submit this assignment:

1. Hand in a physical copy of your answers at the start of class. This is probably the easiest way to submit if you are on campus.
2. Submit a physical copy of your answers in the filing cabinet in the open space near the handout hangout in the Gates building. If you haven't been there before, it's right inside the entrance labeled “Stanford Engineering Venture Fund Laboratories.” There will be a clearly-labeled filing cabinet where you can submit your solutions.
3. Send an email with an electronic copy of your answers to the submission mailing list (cs103-aut1213-submissions@lists.stanford.edu) with the string “[PS2]” somewhere in the subject line.

Extra Credit Problem: Egyptian Fractions (5 Points Extra Credit)

The Fibonacci sequence 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.

In lecture, we saw a surprising connection between Fibonacci numbers and *continued fractions*, an system for writing out rational numbers. Interestingly, *Liber Abaci* also described a separate notation for fractions called *Egyptian fractions*, a method for writing out fractions that has been employed since ancient times.* An Egyptian Fraction is a sum of <u>distinct</u> fractions whose numerators are all one (these fractions are called *unit fractions*). For example:

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

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

* There is archaeological evidence (the Rhind papyrus) that shows that the ancient Egyptians were using Egyptian fractions over three thousand years ago.

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 42 / 137, we would start off by noting that 1 / 4 is the largest unit fraction less than 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 31 / 548 and subtracting it out. This number is 1/18, so we get

$$\frac{42}{137}=\frac{1}{4}+\left(\frac{42}{137}-\frac{1}{4}\right)=\frac{1}{4}+\frac{1}{18}+\left(\frac{31}{548}-\frac{1}{18}\right)=\frac{1}{4}+\frac{1}{18}+\frac{5}{4{,}932}$$

The largest unit fraction we can subtract from 5 / 4932 is 1 / 987:

$$\frac{42}{137}=\frac{1}{4}+\frac{1}{18}+\left(\frac{5}{4{,}932}-\frac{1}{987}\right)=\frac{1}{4}+\frac{1}{18}+\frac{1}{987}+\frac{1}{1{,}622{,}628}$$

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

Prove that the greedy algorithm for continued fractions always terminates for any rational number q in the range $0 < q < 1$ and always produces a valid Egyptian fraction. That is, the sum of the unit fractions should be the original number, and no unit fraction should be repeated. This shows that every rational number in the range $0 < q < 1$ has at least one Egyptian fraction representation.