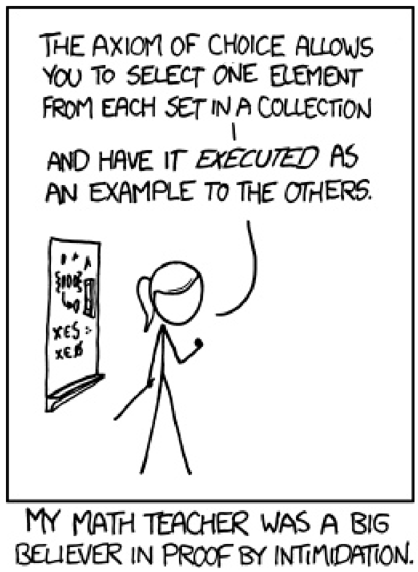
CS302 - Assignment 1

Due: Thursday, Feb. 16 at the beginning of class
Hand-in method: paper



http://xkcd.com/982/

For this assignment you *must* use latex to generate your work. I've included the latex for this handout on the web page if you want to use it as a starting point[1] and I've also included the latex for the notes from the first day of class if you'd like to look at the examples there.

You can use your favorite text editor in combination with `pdflatex` on the command-line or you can install an editor with a built-in latex compiler such as Miktek for Windows or TeXShop on Macs. If you need help with this, please come see me. I've put some latex tutorials on the course web page, but I also encourage you to search the web if you have particular questions.

1. [**5 points**] Pick two different sorting functions from a programming language (e.g. `java.util.Arrays.sort`) and state what sorting algorithm is used. They could be two different ways of sorting in the same language or two different languages. Make sure to cite your source. Why do you think these choices were made?

2. [**5 points**] Proofs

[1] I removed the code that puts in the comic, though

To get you warmed up, here is an example proof showing that $\log_b xy = \log_b x + \log_b y$.

Let:

$k = \log_b xy$
$\ell = \log_b x$
$m = \log_b y$

We want to show that $k = \ell + m$.

- By the definition of logarithms, we know:

$b^k = xy$
$b^\ell = x$
$b^m = y$

- From these, by properties of exponents

$b^k = b^\ell b^m = b^{\ell+m}$

Taking the log of both sides, we obtain $k = \ell + m$, which is what we wanted to show.

Now, give a proof showing:

$\log_b a^n = n \log_b a$

Make sure your proof is clear.

3. [**8 points**] Inductive proofs

   (a) Prove using induction that for all $n \geq 0$:

   $$1 + a + a^2 + a^3 + \ldots + a^n = \frac{a^{n+1} - 1}{a - 1}$$

   where a is any number not equal to 1. Like the example in class, make sure to make clear the base case and the different components of the inductive case.

   (b) Where in your proof did you require that $a \neq 1$?

   (c) What does the sum evaluate to when $a = 1$?

4. Loops

   (a) [**5 points**] Write *pseudocode* for a function `sum_positive` that takes as input an array of numbers and returns the sum of all elements in that array that are positive.

   (b) [**7 points**] Prove that your function is correct by stating a loop invariant and then proving that the invariant is true.