CS 124 Math Review Section

CS 124 is more math intensive than most of the introductory courses in the department. You're going to need to be able to do two things:

1. Perform some clever calculations to understand how your algorithms behave.
2. Use clear mathematical language to describe your algorithms to other people (and to convince yourself that they do what you think they do).

Number 1 is something all of you are going to be able to do, but I'll start off by reviewing some material from calculus, probability, etc. that you might find particularly useful in algorithmic analysis.

Number 2 is trickier. Only the crystal-clear language of mathematical proof will convince your skeptical TFs that you know what you're doing. Some of you might not have written a (good) mathematical proof before. Most of our time will be spent going over how to do that.

1 Some calculation techniques

1.1 Probability and counting

Professor Mitzenmacher threw-around some probability theory in the first class. Here's a problem, important enough that we'll return to it later in the semester, that you should be able to do solve now:

Problem 1 (Probability and expected value): Suppose that I have 17 bins in a row in front of me and I'm throwing a bunch of balls towards them. For each ball, there's a 1/5 chance that it lands in the bin on the far left and a 1/20 chance that it lands in each of the other bins.

- I throw two balls. What is the probability that both land in the bin on the far left? What is the probability that neither lands in the bin on the far right?
- Let $c = 0$. From left to right, I number the bins $1, 2, \ldots, 17$. I throw 100 balls at the bins. For each ball, I figure out what bin it's in, and I add the bin's number to c. What is the expected value of c?
- I throw 5 balls. What is the chance that one will land in the bin on the far left and no two balls will land in the same bin?

1.2 Sums and series

Being able to quickly calculate

$$\sum_{i=1}^{n} i$$

saved us a lot of time on the second part of Problem 1. It often happens happens in CS that we are given a sequence of numbers and need to add them together. If the sequence has a pattern, we can exploit the pattern to avoid adding the numbers one at a time. Here's another example:

$$\sum_{i=0}^{n} (1/2)^i$$

Problem 2: What's the value of this sum in terms of n?

The above two expressions are examples of finite sums. You'll remember from calculus that we can go a step further and consider an infinite series:

$$\lim_{n\to\infty} \sum_{i=0}^{n} .5^i = \sum_{i=0}^{\infty} .5^i$$

All we're asking here is "what happens to the value we found in Problem 2 when n gets really big?". Often, the sum just blows-up to infinity. But in the given case, the numbers that we're adding get very very small so the sum is not infinite (recall Zeno's paradox).

Problem 3 What is the value of the above sum?

Be careful. Just because the terms in a sum get small doesn't mean that the sum converges:

$$\sum_{i=0}^{\infty} 1/i = \infty$$

Dealing with tricky sums is a great topic for a real analysis course. In this course we'll just use sums for our own ends. Appendix A in CLRS has a huge collection of useful sums. Two in particular:

$$\sum_{n=0}^{\infty} r^n = \begin{cases} \frac{1}{1-r} & \text{if } 0 < r < 1, \\ \infty & \text{if } r \geq 1. \end{cases}$$

$$\sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x$$

2 How to write a proof

A proof really isn't anything fancy, it's just a rock-solid mathematical argument. You write a proof to convince the other guy that you're correct. But the point of a proof is more than that. I could just say "I'm the TF and I know what I'm talking about" and you'd probably decide that I'm correct. A proof on the other hand leads the reader step by step from what they already believe to what you want them to believe. They should feel like they're being gently carried along to a higher state of knowledge. This requires a couple things:

1. Start with what the reader already believes.
2. Proceed in logical steps, building on what you've already proved. Don't obsess over every statement that you make, just ensure that each statement follows logically from the statement before it. Make sure that your language is clear and straightforward.
3. ????
4. Profit

That's all there is to it. It's tempting to just write down whatever your train of thought was when you solved the problem. Don't. Write something that is clear, direct, and pleasant to read. Solving the problem is just the first step. There are many paths to a solution and a good proof takes the most beautiful one.

2.1 Important things to not do

Do not substitute an *example* for a proof. Do not substitute a *picture* for a proof. Do not skip logical steps because you're having trouble writing what you mean. There is a very good chance that you are having trouble because you misunderstand something. Suppose that I ask you to prove that

$$F_n = 2^{2^n} + 1$$

is prime for all n. It's easy to check that this is true for $n = 0, 1, 2, 3$, and 4 but hard to prove in general (because it's wrong). Make sure that you're proving all of what you're asked to prove.

Bonus: Why are the "Fermat primes" useful in computer science? (We'll come back to this later in the semester).

2.2 Non-examples and an example

Example problem: We have a biased coin that gives heads with probability q. I have a genius strategy to produce unbiased coin flips using this biased coin. I start by flipping the biased coin twice. I look at these two biased flips. If they are "HT," I add "H" to the output stream. If they are "TH", I add "T" to the output string. Then I flip the coin again, look at the second and third flips, and do the same thing. Then I flip again and look at the third and fourth flips. And so on. If the pair of input flips that I look at is "HH" or "TT" then I skip this pair completely, flipping the biased coin twice in a row in order to get a new pair. Prove that the coin flips in the output stream are biased.

Bad (1/10): Two output flips that I produce with this method are related to each other because they both share one of the same input flips. The fact that they share an input flip means that one of them depends on the other so that the output flips are not unbiased.

Why it's bad: This is the correct idea, but the presentation is shoddy. The remark "they both share one of the same input flips" is unacceptably vague, even though we can figure out from context what it means. The conclusion of the argument is "one of them depends on the other" but I, as the reader, don't understand how the word "depends" relates to the mathematical idea of being "biased".

Better but still bad (6/10): If a stream of coin flips is unbiased, then the probability that the n^{th} flip is heads does not depend on the outcome of the first $n-1$ flips. Suppose that we use the above method to generate a stream of coin flips. If the k^{th} flip in the output stream is "H" then the $(k+1)^{th}$ flip in the output stream is more likely to be a "T" than an "H". To see this, consider what happens when we are producing these output flips. We see "HT" and produce "H" as the k^{th} output flip. Then the next input bit is either "T" or "H". If it is "T" then we start over. If it is "H" then we produce "T". Thus we are more likely to produce a "T" next.

Why it's bad: This is much better. This person states clearly what they mean by "unbiased" and states clearly why this stream is not unbiased. However, they trail off at the end. They have

shown one way that you might get an "H" and then get a "T". They have not proven that an "H" is more likely to be followed by a "T" than by an "H".

Acceptable: If a stream of coin flips is unbiased, then the probability that the n^{th} flip is heads does not depend on the outcome of the first $n-1$ flips. Suppose that we use the above method to generate a stream of coin flips. If the k^{th} flip in the output stream is "H" then the $(k+1)^{th}$ flip in the output stream is more likely to be a "T" than an "H". To see this, consider what happens when we are producing these output flips. We see "HT" and produce "H" as the k^{th} output flip. With probability $1-q$, the next input flip is "H" and we produce a "T". With probability q, the next input flip is "T" and there is equal probability of the next output flip being "H" and "T". This proves the result.

2.3 Proof techniques

2.3.1 Proof by induction

We often use induction when we want to prove that a certain statement is correct for any natural number n. This seems like a tricksy thing to do. If I prove correctness for $n = 0$ and $n = 1$ and $n = 2$, I will have failed CS 124 long before I've proved the statement for $F_5 = 4294967297$. So here's the clever hack.

Establish a **base case**: prove that the statement is true for the minimum value of n that I am interested in (usually $n = 0$ or $n = 1$).

Now fire up the **induction machine**: supposing that the statement is true for $n = k$ (k is a dummy variable), prove that the statement is true for $n = k + 1$.

The induction machine now chews through all of the natural numbers for us.

Another way of phrasing it: if you want to climb up a ladder then you only need two things: You need to be on a rung of the ladder. You need to know how to get from a rung to the rung above it. If you know these two things, you can climb the whole ladder.

When you establish a base case, you prove that you are on a rung of the ladder. When you prove that you have a working induction machine, you prove that you have a gizmo that gets you from one rung to the next.

Example proof: What is the sum of the first n odd numbers?

Proof We will prove by induction that the sum of the first n odd numbers is n^2.

The sum of the first 1 odd numbers is 1, so our claim is valid when $n = 1$.

Suppose that the sum of the first k odd numbers is k^2. Consider the sum of the first $k+1$ odd numbers:

$$1 + (2 \cdot 2 - 1) + \ldots + (2 \cdot k - 1) + (2 \cdot (k+1) - 1)$$

By the induction hypothesis, this is equal to

$$k^2 + (2 \cdot (k+1) - 1) = k^2 + 2k + 1 = (k+1)^2$$

The result follows by induction. □
That's all there is to it! Give it a try yourself:

Problem 4: The triangle inequality says that for x, y real numbers: $|x| + |y| \geq |x + y|$. Prove the following for any n: if $x_1, \ldots, x_n$ are real numbers then $\sum |x_i| \geq |\sum x_i|$.

Problem 5: Prove an exercise from the class notes for the second lecture: $F_n \geq 2^{n/2}$ for $n \geq 8$. (Where F_n is the n^{th} Fibonacci number.)

2.3.2 Proof by contradiction

Prime numbers are fantastically useful in computer science (anyone know an example of why?). Fortunately for us, there are infinitely many prime numbers.

How do we prove something like this. It seems too vague to get a good handle on. Well, let's pretend for a minute that it's false (pretending that the answer is false is often a good way to figure out why it's true, even if we don't end up using a proof by contradiction).

If the statement is false then there are only n prime numbers: $p_1, \ldots, p_n$. Here's a proof by contradiction:

Proof Suppose that the only prime numbers are $p_1, \ldots, p_n$. Consider the number $N = p_1 \cdot \ldots \cdot p_n + 1$. Either N is prime or it is divisible by a prime number. It is larger than any of the p_i, so it must not be prime. Thus it has a prime divisor. However, none of the p_i can divide N (because they divide $N - 1$), so the p_i are not all of the primes. This is a contradiction. There must be infinitely many primes. □

Easy peasy. Try it yourself:

Problem 6: Prove that $\sqrt{2}$ is irrational.