Week 1 - Wednesday

COMP 4500

Last time

- Course overview
- Big Theta of code

Questions?

Assignment 1

Logical warmup

- You come to a fork in the road
- Two men stand beneath a sign that reads:
  - Ask for the way, but waste not your breath
  - One road is freedom, the other is death
  - Just one of the pair will lead you aright
  - For one is a Knave, the other a Knight
- What single yes-or-no question can you ask to determine which fork to take?

Back to Big Theta

Steps on steps on steps

- What's the Big Theta bound if *n* is **`n`**?

```
int counter = 1;
for(int i = 1; i <= n; ++i) {
    for(int j = 0; j < counter; ++j)
        System.out.println("$");
    counter *= 2;
}
```

Switch it up

- What's the Big Theta bound if *n* is **`n`**?

```
int counter = 1;
for(int i = 1; i <= n; ++i) {
    for(int j = 0; j < n/counter; ++j)
        System.out.println("$");
    counter *= 2;
}
```

Back to your roots

- What's the Big Theta bound if *n* is **`n`**?

```
for(int i = 0; i*i < n; ++i)
    for(int j = 0; j < n; ++j)
        System.out.println("%");
```

Finding the Big Theta of loops

- For difficult loops, there are two challenges:
  1. Turning the loops into summation notation
  2. Simplifying the summation into closed-form expressions (without the Σ)
- Practice both parts!

Proofs

The nature of a proof

- A proof is a tool to convince ourselves (and others) that a statement is completely true
- A direct proof starts with a set of true statements:
  - Axioms (things that are always true)
  - Premises (things that we assume are true for this proof)
- Then, you take those true things and generate more true statements using definitions, the laws of mathematics, and logic
- When you're able to generate the conclusion you wanted to prove, you're done!

Universal quantification

- The universal quantifier $\forall$ means "for all"
- The statement "All DJs are mad ill" can be written more formally as:
- $\forall \boldsymbol{x} \in \boldsymbol{D}, \boldsymbol{M}(\boldsymbol{x})$
  - Where $\boldsymbol{D}$ is the set of DJs and $\boldsymbol{M}(\boldsymbol{x})$ denotes that $\boldsymbol{x}$ is mad ill
- We will often want to prove that if something has some property, it will have some other property
- For example:
  - $\forall \boldsymbol{x} \in \boldsymbol{D}, \boldsymbol{B}(\boldsymbol{x}) \rightarrow \boldsymbol{S}(\boldsymbol{x})$
  - Imagine that $\boldsymbol{B}(\boldsymbol{x})$ means that $\boldsymbol{x}$ breaks it down funky style and that $\boldsymbol{S}(\boldsymbol{x})$ means that $\boldsymbol{x}$ stacks cheddar

Existential quantification

- The existential quantifier $\exists$ means "there exists"
- The statement "Some emcee can bust a rhyme" can be written more formally as:
- $\exists \boldsymbol{y} \in \boldsymbol{E}, \boldsymbol{B}(\boldsymbol{y})$
  - Where $\boldsymbol{E}$ is the set of emcees and $\boldsymbol{B}(\boldsymbol{y})$ denotes that $\boldsymbol{y}$ can bust a rhyme

Negating quantified statements

- When doing a negation, negate the predicate and change the universal quantifier to existential or vice versa
- Formally:
  - $\sim(\forall \boldsymbol{x}, \boldsymbol{P}(\boldsymbol{x})) \equiv \exists \boldsymbol{x}, \sim\boldsymbol{P}(\boldsymbol{x})$
  - $\sim(\exists \boldsymbol{x}, \boldsymbol{P}(\boldsymbol{x})) \equiv \forall \boldsymbol{x}, \sim\boldsymbol{P}(\boldsymbol{x})$
- Thus, the negation of "Every dragon breathes fire" is "There is a dragon that does not breathe fire"

Proving Existential Statements and Disproving Universal Ones

Proving existential statements

- A statement like the following:

  $\exists \boldsymbol{x} \in \boldsymbol{D}$ such that $\boldsymbol{P}(\boldsymbol{x})$

- is true, if and only if, you can find at least one element of ***D*** that makes ***P***(***x***) true
- To prove this, you either have to find such an ***x*** or give a set of steps to find one
- Doing so is called a **constructive proof of existence**
- There are also **nonconstructive proofs of existence** that depend on using some other axiom or theorem

Examples

- Prove that there is a positive integer that can be written as the sum of the squares of two positive integers in two distinct ways
- More formally, prove:
  - $\exists \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{a}, \boldsymbol{b} \in \mathbf{Z}^+$ such that $\boldsymbol{x} = \boldsymbol{y}^2 + \boldsymbol{z}^2$ and $\boldsymbol{x} = \boldsymbol{a}^2 + \boldsymbol{b}^2$ and $\boldsymbol{y} \neq \boldsymbol{a}$ and $\boldsymbol{y} \neq \boldsymbol{b}$
- Suppose that ***r*** and ***s*** are integers. Prove that there is an integer ***k*** such that $22\boldsymbol{r} + 18\boldsymbol{s} = 2\boldsymbol{k}$

Disproving universal statements

- Disproving universal statements is structurally similar to proving existential ones
- Instead of needing any single example that works, we need a single example that doesn't work, called a **counterexample**
- Why?
- To disprove $\forall \boldsymbol{x} \in \boldsymbol{D}, \boldsymbol{P}(\boldsymbol{x}) \rightarrow \boldsymbol{Q}(\boldsymbol{x})$, we need to find an $\boldsymbol{x}$ that makes $\boldsymbol{P}(\boldsymbol{x})$ true and $\boldsymbol{Q}(\boldsymbol{x})$ false

Examples

- Using counterexamples, disprove the following statements:
- $\forall \boldsymbol{a}, \boldsymbol{b} \in \mathbf{R}$, if $\boldsymbol{a}^2 = \boldsymbol{b}^2$ then $\boldsymbol{a} = \boldsymbol{b}$
- $\forall \boldsymbol{x} \in \mathbf{Z}$, if $\boldsymbol{x} \geq 2$ and $\boldsymbol{x}$ is odd, $\boldsymbol{x}$ is prime
- $\forall \boldsymbol{y} \in \mathbf{Z}^+$, if $\boldsymbol{y}$ is odd, then $(\boldsymbol{y} - 1)/2$ is prime

Proving Universal Statements

Method of exhaustion

- If the domain is finite, try every possible value
- Example:
  - $\forall \boldsymbol{x} \in \mathbf{Z}^+$, if $4 \leq \boldsymbol{x} \leq 10$ and $\boldsymbol{x}$ is even, $\boldsymbol{x}$ can be written as the sum of two prime numbers
- Is this familiar to anyone?
- Goldbach's Conjecture proposes that this is true for **all** even integers greater than 2

A useful definition

- We'll start with basic definitions of even and odd to allow us to prove simple theorems
- If n is an integer, then:
  - n is even $\Leftrightarrow \exists\, k \in \mathbf{Z}$ such that $n = 2k$
  - n is odd $\Leftrightarrow \exists\, k \in \mathbf{Z}$ such that $n = 2k + 1$
- Since these are bidirectional, each side implies the other

Generalizing from the generic particular

- Pick some specific (but arbitrary) element from the domain
- Show that the property holds for that element, just because of that properties that any such element must have
- Thus, it must be true for all elements with the property
- Example: $\forall \boldsymbol{x} \in \mathbf{Z}$, if $\boldsymbol{x}$ is even, then $\boldsymbol{x} + 1$ is odd

Direct proof

- Direct proof uses the method of generalizing from a generic particular, following these steps:
  1. Express the statement to be proved in the form $\forall \boldsymbol{x} \in \boldsymbol{D}$, if $\boldsymbol{P}(\boldsymbol{x})$ then $\boldsymbol{Q}(\boldsymbol{x})$
  2. Suppose that $\boldsymbol{x}$ is some specific (but arbitrarily chosen) element of $\boldsymbol{D}$ for which $\boldsymbol{P}(\boldsymbol{x})$ is true
  3. Show that the conclusion $\boldsymbol{Q}(\boldsymbol{x})$ is true by using definitions, other theorems, and the rules for logical inference

Direct proof example

- Prove that the sum of any two odd integers is even

Proof by contradiction

- In a proof by contradiction, you begin by assuming the **negation** of the conclusion
- Then, you show that doing so leads to a logical impossibility
- Thus, the assumption must be false and the conclusion true

Contradiction formatting

- A proof by contradiction is different from a direct proof because you are **trying** to get to a point where things don't make sense
- You should always clearly state that it's a **proof by contradiction**
- You will reach a point where you have ***p*** and ~***p***, mark that as a **contradiction**
- If you're doing a proof by contradiction and you actually show the thing you wanted to prove in the first place, **it's not a proof!**

Proof by contradiction example

- **Theorem:** There is no integer that is both even and odd
- **Proof by contradiction:** Assume that there is an integer that is both even and odd

$\sqrt{2}$ is irrational

Theorem: $\sqrt{2}$ is irrational

Proof by contradiction:

1. Suppose $\sqrt{2}$ is rational	1. Negation of conclusion
2. $\sqrt{2} = \boldsymbol{m}/\boldsymbol{n}$, where $\boldsymbol{m},\boldsymbol{n} \in \mathbf{Z}$, $\boldsymbol{n} \neq 0$ and $\boldsymbol{m}$ and $\boldsymbol{n}$ have no common factors	2. Definition of rational
3. $2 = \boldsymbol{m}^2/\boldsymbol{n}^2$	3. Squaring both sides
4. $2\boldsymbol{n}^2 = \boldsymbol{m}^2$	4. Multiply both sides by $\boldsymbol{n}^2$
5. $2\boldsymbol{k} = \boldsymbol{m}^2$, $\boldsymbol{k} \in \mathbf{Z}$	5. Square of integer is integer
6. $\boldsymbol{m} = 2\boldsymbol{a}$, $\boldsymbol{a} \in \mathbf{Z}$	6. Even $\boldsymbol{x}^2$ implies even $\boldsymbol{x}$ (Proven elsewhere)
7. $2\boldsymbol{n}^2 = (2\boldsymbol{a})^2 = 4\boldsymbol{a}^2$	7. Substitution
8. $\boldsymbol{n}^2 = 2\boldsymbol{a}^2$	8. Transitivity
9. $\boldsymbol{n} = 2\boldsymbol{b}$, $\boldsymbol{b} \in \mathbf{Z}$	9. Even $\boldsymbol{x}^2$ implies even $\boldsymbol{x}$
10. 2 divides ***m*** and 2 divides ***n***	10. Conjunction of 6 and 9, **contradiction**
11. $\sqrt{2}$ is irrational	11. By contradiction in 10, supposition is false

■

Three-sentence Summary of Computational Tractability and Asymptotic Orders of Growth

Computational Tractability

Why polynomial time?

- Algorithm designers often consider any algorithm that runs in polynomial time to be "efficient"
  - Obviously untrue for n^{100}
- In practice, most polynomial time algorithms have reasonable exponents
  - And few non-polynomial algorithms run in reasonable time
- All polynomial running times have the property that doubling the input size will increase the work by some constant tied to the highest degree of the polynomial
  - Doubling in a quadratic takes 4 times as much work
  - Doubling in a cubic takes 8 times as much work

Table of running times

Time to do the number of instructions given based on a machine that can do one million instructions per second

	n	$n \log n$	n^2	n^3	1.5^n	2^n	$n!$
10	< 1 s	< 1 s	< 1 s	< 1 s	< 1 s	< 1 s	4 s
30	< 1 s	< 1 s	< 1 s	< 1 s	< 1 s	18 minutes	10^{25} years
50	< 1 s	< 1 s	< 1 s	< 1 s	11 minutes	36 years	∞
100	< 1 s	< 1 s	< 1 s	1 s	12,892 years	10^{17} years	∞
1,000	< 1 s	< 1 s	1 s	18 minutes	∞	∞	∞
10,000	< 1 s	< 1 s	2 minutes	12 days	∞	∞	∞
100,000	< 1 s	2 s	3 hours	32 years	∞	∞	∞
1,000,000	1 s	20 s	12 days	31,710 years	∞	∞	∞

For the purposes of this table, we will mark any value greater than 10^{25} years with ∞. Note that the age of the universe is less than 1.4×10^{10} years

Upcoming

Next time...

- Stable Marriage
- Five representative problems:
  - Interval scheduling
  - Weighted interval scheduling
  - Bipartite matching
  - Independent set
  - Competitive facility location

Reminders

- Read Sections 1.1 and 1.2
- Assignment 1 is due next Friday