Objectives

- Introduction to Algorithms, Analysis
- Course summary
- Reviewing proof techniques

What This Course Is About

From *30 Rock*

> Now, everything comes down to expert knowledge of **algorithms** and **data structures**. If you don't speak fluent **O-notation**, you may have trouble getting your next job at the technology companies in the forefront.
>
> -- Larry Freeman

> For me, great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a **brilliant new light** on some aspect of computing.
>
> -- *Francis Sullivan*

What is an Algorithm?

Questions to Consider

- What are our goals when designing algorithms?
- How do we know when we've met our goals?

Course Goals

- Learn how to formulate precise problem descriptions
- Learn specific algorithm design techniques and how to apply them
- Learn how to analyze algorithms for efficiency and for correctness
- Learn when no exact, efficient solution is possible

Course Content

- Algorithm analysis
  - Formal – proofs; Asymptotic bounds
- Advanced data structures, e.g., heaps, graphs
- Greedy Algorithms
- Dynamic Programming
- Divide and Conquer
- Network Flow
- Computational Intractability

Course Notes

- Textbook: *Algorithm Design*
  - Optional: CLRS
- Participation is encouraged
  - Individual, group, class
- Assignments:
  - Solutions to problems
  - Analysis of solutions
  - Programming

Course Grading

- 40% Individual written and programming homework assignments
- 10% Quizzes
- 25% Midterms
- 20% Final
- 5% Participation and attendance

ALGORITHMS

Computational Problem Solving 101

- Computational Problem
  - A problem that can be solved by logic
- To solve the problem:
  1. Create a model of the problem
  2. Design an algorithm for solving the problem using the model
  3. Write a program that implements the algorithm

Computational Problem Solving 101

- Algorithm: a well-defined recipe for solving a problem
  - Has a finite number of steps
  - Completes in a finite amount of time
- Program
  - An algorithm written in a programming language

PROOFS

Why Proofs?

- What are insufficient alternatives?
- How can we prove something isn't true?

Why Proofs?

- What are insufficient alternatives?
  - Examples
    - Considered all possible?
  - Empirical/statistical evidence
    - Ex: "Lying" with statistics
- How can we prove something isn't true?
  - One counterexample

Need irrefutable proof that something is true—for ***all*** possibilities

Common Types of Proofs

Common Types of Proofs

- Proof by contradiction
- Proof by induction

Proof By Contradiction

What are the steps to a proof by contradiction?

Proof By Contradiction

1. Assume the thing we want to prove is false
2. Reason to a contradiction
3. Conclude that it must therefore be true

Example: There are Infinitely Many Primes

- What is a prime number?
- What is not-a-prime number?

- Assume there are only finitely many prime numbers
  - List them: $p_1, p_2 \ldots, p_n$

Example: There are Infinitely Many Primes

- Assume there are only finitely many prime numbers
  - List them: $p_1, p_2 \ldots, p_n$
- Consider the number $q = p_1p_2\ldots p_n + 1$

What are the possibilities for q?

q is either composite or prime

Example: There are Infinitely Many Primes

- Assume there are only finitely many prime numbers
  - List them: $p_1, p_2 \ldots, p_n$
- Consider the number $q = p_1p_2\ldots p_n + 1$
- Case: q is composite
  - If we divide q by any of the primes, we get a remainder of 1 → q is not composite

Example: There are Infinitely Many Primes

- Assume there are only finitely many prime numbers
  - List them: $p_1, p_2 \ldots, p_n$
- Consider the number $q = p_1p_2\ldots p_n + 1$
- Case: q is composite
  - If we divide q by any of the primes, we get a remainder of 1 → q is not composite
- Therefore, q is prime, but q is larger than any of the finitely enumerated prime numbers listed → *Contradiction*

Proof thanks to Euclid

Proof By Induction

What are the steps to a proof by induction?

Proof By Induction

1. What you want to prove
2. Base case
   - Typical: Show statement holds for $n = 0$ or $n = 1$
3. Assumption for n (***induction hypothesis***)
4. Induction step: show that adding one to n also holds true
   - Often relies on earlier assumptions

When/why is induction useful?

Show true for all (infinite) possibilities
Show works for "one more"

Warm Up

Prove:
$2+4+6+8+\ldots + 2n = n*(n+1)$

Proof

Prove: $2+4+6+8+\ldots + 2n = n*(n+1)$

- **Base case**: $n = 1 \rightarrow 2*1 = 1*(1+1)$ ✔
- Assume true for n
- Prove for $n+1$
  - $2+4+6+8+\ldots + 2n + 2(n+1)$
  - $= n*(n+1) + 2(n+1)$
  - $= n^2 + n + 2n + 1 = n^2 + 3n + 1$
  - $= (n+1)*(n+2)$
  - $= (n+1)*((n+1)+1)$ ✔

Proof: All Horses Are The Same Color

- **Base case**: If there is only *one* horse, there is only one color.
- **Induction step**: Assume as induction hypothesis that within any set of n horses, there is only one color.
  - Look at any set of $n + 1$ horses
  - Label the horses: 1, 2, 3, ..., n, $n + 1$
  - Consider the sets {1, 2, 3, ..., n} and {2, 3, 4, ..., $n + 1$}
  - Each is a set of only n horses, therefore within each there is only one color
  - Since the two sets overlap, there must be only one color among all $n + 1$ horses

Where is the error in the proof?

Error in Proof

- **Base case**: If there is only *one* horse, there is only one color.
- **Induction step**: Assume as induction hypothesis that within any set of n horses, there is only one color.
  - Look at any set of $n + 1$ horses
  - Number them: 1, 2, 3, ..., n, $n + 1$
  - Consider the sets {1, 2, 3, ..., n} and {2, 3, 4, ..., $n + 1$}
  - Each is a set of only n horses, therefore within each there is only one color
  - *Since the two sets overlap*, there must be only one color among all $n + 1$ horses

Does not hold true when n+1=2

Lesson: check assumptions within proof

Proof Summary

- Need to prove conjectures
- Common types of proofs
  - Contradiction
  - Induction
- Common error: not checking/proving assumptions
  - "Jumps" in logic

For Next Time

- Read first two pages of book's preface
- Read Chapter 1 of book