Lecture Review

- Look at the study bees at the end of each lecture.
- The purpose of recitation is to highlight important points from lecture, and help you understand class material. We will not cover everything. If you have questions, ask!

Lecture 1: Pancakes with a Problem

**Bracketing:** zeroes in on the runtime of an algorithm, starting with bounds you can prove off the top of your head and then refining the analysis. If you’re working on a hard problem and can’t prove the main result, don’t let your brain turn off! Sneak up on the answer with easy results from above and below, and catch it in the middle.

- To bound from above, *give an algorithm*.
- To bound from below, *find something that every valid algorithm must compute*.
- To get an exact result, *make your upper and lower bounds equal*.

- Example from lecture: Pancake numbers. \( n - 1 \leq P_n \leq 2n - 1 \).

Lecture 2: Inductive Reasoning: One Step At a Time

**Induction:** will never, ever go away. Know its many faces:

- **Normal.** Base case, Inductive Hypothesis, Inductive step.
- **All previous** (strong). When you need many cases in the IH, or you don’t know which cases you will need. Example: Factorization.
- **Least counterexample** (infinite descent). *Assume for contradiction the statement is false, then there is a least counterexample. Construct a smaller counterexample. Contradiction.*
- **Invariant.** The invariant started out true; every step maintains truth value. Good for proving correctness of algorithms.
- **Structural.** IH: *Assume true for all ”smaller” structures.* Good for graphs and complicated data structures. Be careful how you define ”smaller.”

Things to look out for when doing induction:

- Make sure you have enough base cases and IH cases. Use strong induction if necessary.
- Make sure your assumptions in the inductive step match your inductive hypothesis.
- The difference between ”assume for an arbitrary \( k \)” and ”assume that for all \( k \)”.
Example Problems

- How long does it take to sort an array of \( n \) integers?
  - an upper bound
  - a lower bound
- A timid sequence is a permutation of 1, 2, \ldots, \( n \) such that for every integer \( k \) in the list, either \( k - 1 \) or \( k + 1 \) must have appeared earlier in the list (except for the first element in the list).
  Find a recurrence for \( T(n) \), the number of timid sequences on 1, 2, \ldots, \( n \).
- Prove that for every simple graph \( G \) such that \( n = |V(G)| = |E(G)| \) and \( n \geq 3 \), \( G \) has a triangle. (Use induction.)

Group Problems

These problems are meant to give you a chance to practice the problem-solving and communication skills needed for the homework. Please work in groups of 3 people. When a problem is solved, make sure everybody in your group understands it, and then tell the TA.

Finding a Recurrence

(a) Alice wants to write a secret message to Bob, but their alphabet only consists of the strings "0" and "01". How many valid messages could Alice write if she wants it to be \( n \) characters long?

Proof by Induction

(a) Prove that for every graph \( G \), either \( G \) or \( G^c \) is connected.

(b) Consider the following recurrence: \( T(0) = 2 \), \( T(1) = 5 \), and \( T(n) = T(n - 1) + 2 \cdot T(n - 2) \), \( \forall n \geq 2 \).
  A (smart) little birdie tells you that \( T(n) = \frac{7 \cdot 2^n + (-1)^n}{3} \), \( \forall n \in \mathbb{N} \) is a solution to the recurrence.
  Prove the little birdie's formula using induction.

Misproof by Induction

What are the fallacies in the following arguments?

(a) **Claim:** Let \( a \) be any positive number. For all positive integers \( n \), we have \( a^{n-1} = 1 \).
   **Proof:** If \( n = 1 \), \( a^{n-1} = a^{1-1} = a^0 = 1 \). And by induction, assuming that the theorem is true for 1, 2, \ldots, \( n \), we have
   \[
   a^{(n+1)-1} = a^n = \frac{a^{n-1} \times a^{n-1}}{a^{n-2}} = \frac{1 \times 1}{1} = 1;
   \]
   so the theorem is true for \( n + 1 \) as well.

(b) **Claim:** Every positive integer \( n \) is an integer power of two.
   **Proof:** We perform induction on \( n \). Base case: \( n = 1 = 2^0 \). For the IH, assume that every positive integer \( m \) where \( m < n \) is an integer power of two. \( n = 2 \cdot n/2 = 2 \cdot 2^k \) for some integer \( k \), by the induction hypothesis. Hence \( n = 2^{k+1} \) where \( k + 1 \) is an integer.