# Week 1: Introductions and Induction

CSC 236:Introduction to the Theory of Computation Summer 2024 Instructor: Lily You will do well if you demonstrate thorough understanding of the course content

# Logistics

- Lectures: May 8<sup>th</sup> ~ Aug. 7<sup>th</sup> (11 total + 2 midterms)
- Tutorials: *Starts next week*. 8~9 pm after lectures

Section	Room	Tutor
5101	BA 2159	Yibin
5102	BA 2139	Soroush
5103	BA 2135 Lawrence	
5104	BA B024	Logan

- Office Hours: *(tentative)* BA 2272, Friday 1~3 pm
- Discussion: Piazza (link available on Quercus)

# **Course Description**

- Prerequisites: CSC165 or equivalent
- Textbooks:
  - Required: David Liu's notes and Vassos Hadzilacos' notes (available on Quercus)
  - Supplementary: The Nature of Computation (Moore, Mertens), Concrete Mathematics (Graham, Knuth, Patashnik), Introduction to Automata Theory, Languages, and computation (Hopcroft, Motwani, Ullman)
- Topics Covered:
  - Mathematical Proofs
  - Discrete Maths (e.g. Combinatorics, Graph Theory)
  - Proof-of-correctness
  - Finite Automaton

Sabadula	Lecture: Week	Content
Scheudle	1: May 6 – May 12	Introduction, Review, and Induction
	2: May 13 - May 19	Combinatorics: the art of counting
Maths Proof	3: May 20 - May 26	Induction with Examples in Combinatorics
Tools and Applications	4: May 27 - June 2	Induction with Examples in Graph Theory
	5: June 3 - June 9	Correctness of Algorithms
Proof-of- correctness	6: June 10 - June 16	Recursive Algorithms
	June 17 - June 23	Midterm 1 (Lectures 1~4)
	June 24 - June 30	
	7: July 1 - July 7	Master Method: analysis of recursive algorithms
	8: July 8 - July 14	Introduction to Formal Languages
	July 15 - July 21	Midterm 2 (Lectures 5~7)
Finite Automaton	9: July 22 - July 28	Equivalence of DFA, NFA, and Regex
	10: July 29 - Aug 4	Pumping Lemma and Limitations of Formal Languages
	11: Aug 5 - Aug 11	Review and Student's Choice

## **Marking Scheme**

- Quizzes [11 total (take best 8); 1% each]
  - available afternoon *before* lecture
  - 4~5 questions

Remark Requests: Fill out form (see syllabus)

- Assignments [6 total (take best 4); 8% each]
  - 2 warm-up questions (unmarked)
  - 2 questions (marked): peer review required
  - Late work: 10% deduction per day and *at most 3 days*
- Midterms [During June exam week and July 17; 15% each]
  - Missed midterm? 30min~1hour oral exam to reassign marks to final
- Final Exam [time TBD; 30%] *minimum 40% to pass*

### **Review: Asymptotic Analysis**



 $f \in O(g)$  and  $f \in \Omega(g) \to f \in \Theta(g)$ 

## **Review: Simple Induction**

In this class induction will be on discrete sets though this is <u>not</u> <u>generally</u> true.

Example. Prove  $(1 + x)^n \ge (1 + nx)$  for  $x \in \mathbb{R}_{\ge 0}$  and  $n \in \mathbb{N}$ .

- 1. Predicate: P(n)
- 2. Base Case: P(0) is true
- 3. Inductive step:  $P(0) \land \forall k, P(k) \rightarrow P(k+1)$

#### 4. Conclusion: $\forall n \in \mathbb{N}$ , P(n) is true.

# Now you try!

Q1. Use *induction* to prove that in a set S with n = |S| elements, there are  $\frac{n(n-1)(n-2)}{6}$  subsets of size exactly three.

Q2. Produce an algorithm which outputs all subsets of size exactly three. Give its tight asymptotic running time analysis.

**Q1.** Use induction to prove that in a set *S* with n = |S| elements, there are  $\frac{n(n-1)(n-2)}{6}$  subsets of size exactly three.

**Q2.** Produce an algorithm which outputs all subsets of size exactly. Give its tight asymptotic running time analysis.

### Strong/Complete Induction

- Example. For all  $n \in \mathbb{N}$  where n > 1, n has a prime factorization.
- 1. Predicate: P(n)
- 2. Base Case: P(0) is true
- 3. Inductive step:  $P(0) \land \forall k, (P(0) \land P(1) \land \dots \land P(k)) \rightarrow P(k+1)$ Simple Induction:  $P(0) \land \forall k, P(k) \rightarrow P(k+1)$

4. Conclusion:  $\forall n \in \mathbb{N}$ , P(n) is true.

### Example. Fibonacci Sequence

$$f_0 = 0, f_1 = 1, f_n = f_{n-1} + f_{n-2}$$
  
Prove using induction that  $f_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}$  for all  $n \in \mathbb{N}$ .

Let  $f_n$  be the *n*th Fibonacci number. Prove that  $f_n \leq (7/4)^{n-1}$ .

Certainly! Let's prove that for all positive integers n, the nth Fibonacci number, denoted as  $f_n$ , satisfies the inequality:  $f_n \leq (7/4)^{n-1}$ . We'll use mathematical induction to prove this. **Base Case**: For n = 1, we have  $f_1 = 1$ , and  $(7/4)^0 = 1$ . So, the base case holds. **Inductive Hypothesis:** Assume that the inequality holds for some positive integer k, i.e.  $f_k \leq 1$  $(7/4)^{k-1}$ **Inductive Step**: We need to show that the inequality holds for k + 1: Starting with the definition of the Fibonacci sequence:  $f_{k+1} = f_k + f_{k-1}$ By our inductive hypothesis :  $f_k \leq (7/4)^{k-1}$ And since Fibonacci numbers are positive, we can multiply both sides of the inequality by  $f_k$ :  $f_k \cdot f_{k-1} \le f_k \cdot (7/4)^{k-1}$ Now, let's use the fact that  $f_k = f_{k-1} + f_{k-2}$ :  $f_{k+1} = f_k + f_{k-1} \le f_k + f_k = 2f_k$ Combining the inequalities:  $2f_k \leq 2(7/4)^{k-1}$ . Finally, we have:  $f_{k+1} \leq 2(7/4)^{k-1} \leq (7/4)^k$ . This completes the inductive step...

# Recap

- Course syllabus
- Recap of prior content
  - asymptotic analysis
    - Definitions of  $O, \Omega, \Theta$  notation
  - simple induction
    - Steps of the process: predicate, base case, inductive step (using IH)
- Strong/complete induction
  - Ex. Prime decomposition, Fibonacci sequence

Next time: theory behind induction and counting