Week 1: Introductions and Induction

CSC 236:Introduction to the Theory of Computation

Summer 2024

Instructor: Lily

think carefully about
<u>how</u> you are representing your knowledge

You will do well if you demonstrate thorough understanding of the course content

→ math proofs
→ correctness of algorithms
→ finite automaton


FRIDEN

Logistics

- Lectures: May 8th ~ Aug. 7th (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

Schedule

Maths Proof Tools and Applications

Proof-of-correctness

Finite Automaton

Lecture: Week	Content
1: May 6 – May 12	Introduction, Review, and Induction
2: May 13 - May 19	Combinatorics: the art of counting
3: May 20 - May 26	Induction with Examples in Combinatorics
4: May 27 - June 2	Induction with Examples in Graph Theory
5: June 3 - June 9	Correctness of Algorithms
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)*
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)

20%

- 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*

IDK

- 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

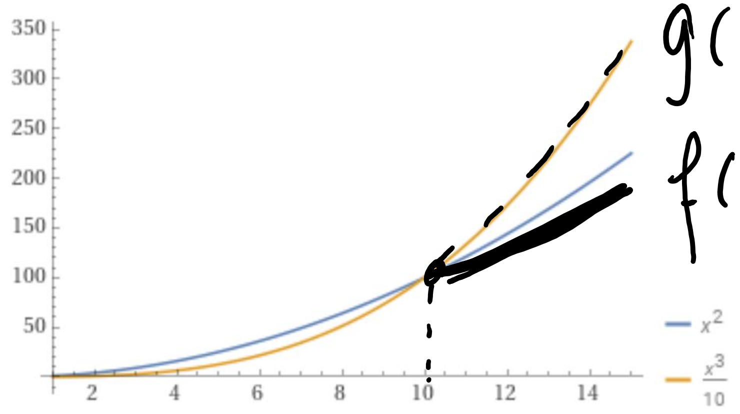
also ok $f = O(g)$ $\quad f,g : \mathbb{Z}_+ \to \mathbb{R}_{\geq 0}$

$f \in O(g)$

$\exists c, \exists n_0, \forall n \geq n_0 : f(n) \leq cg(n)$

$f \in \Omega(g)$

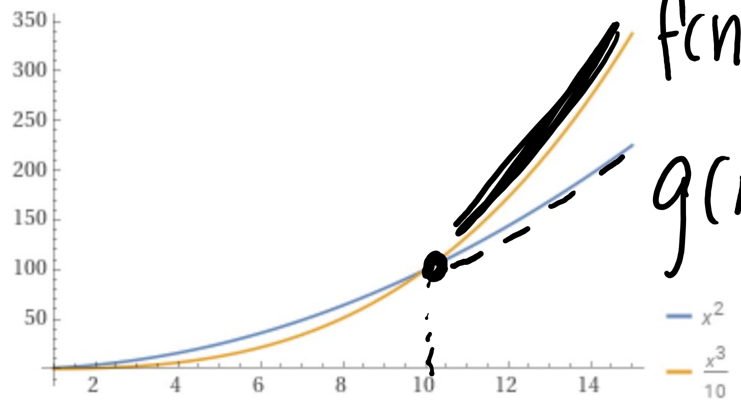
$\exists c, \exists n_0, \forall n \geq n_0 : f(n) \geq cg(n)$



$g(n) = n^3/10$

$f(n) = n^2$

$n_0 \longrightarrow$



$f(n) = n^3/10$

$g(n)$ n^2

$n_0 \longrightarrow$

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

Review: Simple Induction

In this class induction will be on discrete sets though this is [not generally]() true.

induct on n

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

1. Predicate: $P(n)$ $:=$ for $x \in \mathbb{R}_{\geq 0}$, $(1+x)^n \geq (1+nx)$
2. Base Case: $P(0)$ is true $\quad n=0 \quad 1 = (1+x)^{n \to 0} \geq (1+\overset{\to 0}{n}x) = 1$
3. Inductive step: $P(0) \wedge \forall k, P(k) \rightarrow P(k+1)$

   assume $P(k)$ is true. show $P(k+1)$

   $$(1+x)^{k+1} = (1+x)^k(1+x) \overset{(IH)}{\geq} (1+kx)(1+x)$$

   $$= 1 + (k+1)x + \underbrace{kx^2}_{\geq 0}$$

   $$\geq 1 + (k+1)x$$
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.

$S = \{1,2,3,4\}$ $\{1,2,3\} = \{1,3,2\}$

$P(n) :=$ a set S with n elements has exactly $\frac{n(n-1)(n-2)}{6}$ sets of exactly 3

$\in O(n^3)$

base case: $n = 0, 1, 2$ $\frac{n(n-1)(n-2)}{6} = 0$

inductive step: suppose $P(k)$ is true $(k \geq 2)$ (IH). Show $P(k+1)$ is true

KNOW $|S| \geq 1$, say $a \in S$ (WLOG) $S \setminus \{a\}$ has $\frac{k(k-1)(k-2)}{6}$ sets of size exactly 3 (i)

set of size 3 containing a: $k(k-1)/2$ (ii)

$k(k-1)\left(\frac{k-2}{6} + \frac{1}{2}\right) = \frac{k(k-1)(k+1)}{6}$

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

```
def find(S: set)
1     result = []
2     for i in range(n):
3         for j in range(i+1, n):
4             for k in range(j+1, n):
5                 result.append((i, j, k))
      return result
```

UB: $O(n^3)$

Lines 2, 3, 4 each run in at most $O(n)$ iterations (and nested)

LB: $\Omega(n^3)$, $S = \{1, \dots, n\}$, $\Omega(n^3)$ on S.

$\Rightarrow T(n) = \Theta(n^3)$

Strong/Complete Induction

$n = p_1 \cdot \dots \cdot p_m$

p_i is prime

Example. For all $n \in \mathbb{N}$ where $n > 1$, n has a prime factorization.

1. Predicate: $P(n)$:= n has prime Factorization.

$k+1 = p_1 \cdots p_m \cdot q_1 \cdots q_l$

2. Base Case: $P(\cancel{0}\,2)$ is true $n = 2$ is prime

3. Inductive step: $P(0) \wedge \forall k, (P(\cancel{0}\,2) \wedge P(1) \wedge \cdots \wedge P(k)) \rightarrow P(k+1)$

Simple Induction: $P(0) \wedge \forall k, P(k) \rightarrow P(k+1)$

fix integer k, assume $P(2) \cdots P(k)$ true, WTS $P(k+1)$ is true.

$k+1$ (1) is prime ✓ → $\exists a : \frac{k+1}{a}$ is integer

4. Conclusion: $\forall n \in \mathbb{N}, P(n)$ is true. (2) is composite

$a, \frac{k+1}{a} < k+1 \Rightarrow a, \frac{k+1}{a}$ have prime fact. $\left.\begin{array}{l} a = p_1 \cdots p_m \\ \frac{k+1}{a} = q_1 \cdots q_l \end{array}\right\}$

Example. Fibonacci Sequence

$\left(\frac{1-\sqrt{5}}{2}\right)^2 = \frac{3-\sqrt{5}}{2}$

$f_0 = 0, f_1 = 1, f_n = f_{n-1} + f_{n-2}$ $\quad f_2 = 1,\ f_3 = 2, \ldots$

Prove using induction that $f_n = \dfrac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}$ for all $n \in \mathbb{N}$. induct on n

base case $n=0,1$ $\quad f_0 = 0,\ f_1 = 1$ $\quad$ for fixed k

inductive step $P(0) \wedge \cdots \wedge P(k+1) \rightarrow P(k+2)$

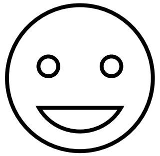
(IH)

$\left(\frac{1+\sqrt{5}}{2}\right)^2 = \frac{3+\sqrt{5}}{2}$

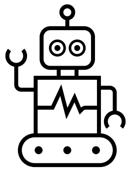
$f_{k+2} = f_{k+1} + f_k = \left(\left(\frac{1+\sqrt{5}}{2}\right)^{k+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{k+1} + \left(\frac{1+\sqrt{5}}{2}\right)^k - \left(\frac{1-\sqrt{5}}{2}\right)^k\right) \cdot \frac{1}{\sqrt{5}}$

$= \left(\left(\frac{1+\sqrt{5}}{2}\right)^k\left(\frac{1+\sqrt{5}}{2}+1\right) - \left(\frac{1-\sqrt{5}}{2}\right)^k\left(\frac{1-\sqrt{5}}{2}+1\right)\right) \cdot \frac{1}{\sqrt{5}}$

$= \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^{k+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{k+2}\right)$



Let f_n be the nth 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.

structure ok!

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. $f_k \leq (7/4)^{k-1}$

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:

$$f_k \cdot f_{k-1} \leq f_k \cdot (7/4)^{k-1}$$

?

Now, let's use the fact that $f_k = f_{k-1} + f_{k-2}$:

$$f_{k+1} = f_k + f_{k-1} \leq f_k + f_k = 2f_k$$

How are these parts related?

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, Ω, Θ 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