# CS4800: Algorithms — S'18 — Jonathan Ullman

Homework 1
Due Friday January 19 at 4:59pm via Gradescope

Name:
Collaborators:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Friday January 19 at 4:59pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.
- Solutions must be typeset in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly prohibited.

**Problem 1.** *Asymptotic Order of Growth*

Rank the following functions in increasing order of asymptotic growth rate. That is, find an ordering $f_1, f_2, \ldots, f_{12}$ of the functions so that $f_i = O(f_{i+1})$. No justification is required.

| | | | | | |
|---|---|---|---|---|---|
| $n^3$ | $7^{\log_2 n}$ | $n!$ | $12^n$ | $\log_2(n!)$ | $2^{4n}$ |
| $(\log_2 n)^2$ | $(\log_2 n)(\log_2 \log_2 n)^2$ | $\sqrt{n}$ | $2^{\log_3 n}$ | $16^{\sqrt{\log_2 n}}$ | $128n$ |

**Solution:**

**Problem 2.** *Inductive Proofs*

(a) Prove the following statement by induction: For every $n \in \mathbb{N}$, $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

**Solution:**

(b) Prove the following statement by induction: For every $n \in \mathbb{N}$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - \frac{1}{n}$

**Solution:**

(c) Your friend shows you the following dubious theorem and proof.

**Theorem 1.** *In every set of $n \geq 1$ lemurs, all lemurs are the same size.*

*Proof.* Base Case ($n = 1$): Because the set has only one lemur, it has the same size as itself.

Inductive Step: Assume that the theorem is true for $n = k$. We will prove the theorem for $n = k + 1$. Consider a set of $k + 1$ lemurs $a_1, \ldots, a_k, a_{k+1}$. By our assumption, the first $k$ lemurs are the same size.

$$\underbrace{a_1, a_2, \ldots, a_k}_{\text{same size}}, a_{k+1}$$

Also by our assumption, the last $k$ lemurs also have the same color.

$$a_1, \underbrace{a_2, \ldots, a_k, a_{k+1}}_{\text{same size}}$$

Therefore, by transitivity, all lemurs in the set are the same size.

Thus, by induction, the theorem is true for all $n \geq 1$. □

What is the bug in this proof?

**Solution:**

**Problem 3.** *Karatsuba Example*

Carry out Karatsuba's Algorithm to compute $14 \cdot 92$. What are the inputs for each recursive call and what does that recursive call return?

**Solution:**

**Problem 4.** *What Does This Code Do?*

You encounter the following mysterious piece of code.

```
function COMPUTE(a, n)
    if n = 0 then
        return 1
    else if n = 1 then
        return a
    else if n even then
        u ←COMPUTE(a, ⌊n/2⌋)
        return u · u
    else if n odd then
        u ←COMPUTE(a, ⌊n/2⌋)
        return a · u · u
    end if
end function
```

(a) What are the results of COMPUTE($a$, 2), COMPUTE($a$, 3), and COMPUTE($a$, 4)? You do not need to justify your answers.

**Solution:**

(b) What does the code do in general? Prove your assertion by induction on $n$.

**Solution:**

(c) Write the recurrence for the running time as a function of $n$ and show $T(n) = O(\log n)$ by induction. Assume that each multiplication takes one unit of time and you only need to count the number of multiplications.

**Solution:**