CS3000: Algorithms & Data — Fall '18 — Jonathan Ullman

Homework 1
Due Tuesday September 18 at 11:59pm via Gradescope

Name: Kevin Zhang
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 Tuesday September 18 at 11:59pm 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 forbidden.

Problem 1. *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 $n \geq 1$ dice, all dice are the same color.*

Proof. **Inductive Hypothesis:** Let $H(k)$ be the statement: in every set of k dice, all of the k dice are the same color. We will prove that $H(k)$ is true for every $k \in \mathbb{N}$.

Base Case: Consider $H(1)$. Because the set has only one die, it is the same color at itself, so $H(1)$ is true.

Inductive Step: We will show that for every $k \geq 1$, $H(k) \implies H(k+1)$. Assume that $H(k)$ is true. Consider a set of $k+1$ dice $d_1, \ldots, d_k, d_{k+1}$. By our assumption, the first k dice are the same color.

$$\underbrace{d_1, d_2, \ldots, d_k}_{\text{same color}}, d_{k+1}$$

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

$$d_1, \underbrace{d_2, \ldots, d_k, d_{k+1}}_{\text{same color}}$$

Therefore, by transitivity, all dice are the same color.

Therefore, the claim holds for all n by induction. $\square$

What is the bug in this proof?

Solution:

Problem 2. *Stable Matching*

In class we showed that given *any* set of rankings for n doctors and n hospitals, there always exists at least one stable matching of doctors and hospitals.

(a) Show that there is a set of rankings for 2 doctors and 2 hospitals such that there is a *unique* stable matching. Justify the claim that there is a unique stable matching.

Solution:

(b) Show that there is a set of rankings for 2 doctors and 2 hospitals such that there exist two distinct stable matchings.

Solution:

(c) Show that, for every n, there is a set of rankings for $2n$ doctors and $2n$ hospitals such that there are at least 2^n distinct stable matchings. *Hint: start with your answer to part (b) and build your ranking two pairs at a time.*

Solution:

Problem 3. *Asymptotic Order of Growth*

(a) 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.

$$\begin{array}{ccccc} n^3 & 7^{\log_2 n} & n! & 12^n & \log_2(n!) \\ 2^{4n} & 100n^{3/2} & 10n & 2^{\log_3 n} & \log_2^3 n \end{array}$$

Solution:

$$f_1(n) = ???$$
$$f_2(n) = ???$$
$$\ldots$$

(b) Suppose $f(n), g(n), h(n)$ are non-decreasing, non-negative functions and that $f(n) = O(h(n))$ and $g(n) = O(h(n))$. Prove that the $f(n)g(n) = O(h(n)^2)$.

Solution:

Problem 4. *What Does This Code Do?*

You encounter the following mysterious piece of code.

Algorithm 1: Mystery Function

```
Function C(a, n):
    If n = 0 :
        Return (1, a)
    ElseIf n = 1 :
        Return (a, a · a)
    ElseIf n is even :
        (u, v) ← C(a, ⌊n/2⌋)
        Return (u · u, u · v)
    ElseIf n is odd :
        (u, v) ← C(a, ⌊n/2⌋)
        Return (u · v, v · v)
```

(a) What are the results of $C(a,2)$, $C(a,3)$, and $C(a,4)$. 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) In this problem you will analyze the running time of C as a function of n. Prove that, for every $n \in \mathbb{N}$, the number of multiplication operations performed in evaluating $C(a,n)$ is at most $2 \cdot \log_2 n$.

Solution: