CS3000: Algorithms & Data — Spring 2019 — Paul Hand

Homework 1
Due Wednesday 1/16/2019 at 2:50pm 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 Wednesday 1/16/2019 at 2:50pm 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. You may use the internet for general research and learning pertaining to class material.

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 $k \geq 1$, $H(k) \Longrightarrow H(k+1)$. 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. □

What is the error in this proof?

Solution:

Problem 2. *Stable Matching*

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

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

Solution:

Here is a placeholder for how to provide a set of preference in LaTeX.

Candidates	Jobs
$c_1 : j_1 \succ j_2 \succ j_3$	$j_1 : c_1 \succ c_2 \succ c_3$
$c_2 : j_1 \succ j_2 \succ j_3$	$j_2 : c_1 \succ c_2 \succ c_3$
$c_3 : j_1 \succ j_2 \succ j_3$	$j_3 : c_1 \succ c_2 \succ c_3$

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

Solution:

(c) Show that, for every n, there is a set of rankings for $2n$ candidates and $2n$ jobs 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. *Stable Matching Variant*

In this problem we will study a variant of the stable matching problem presented in class. Consider a room of n students which are to form pairs in order to complete a homework assignment. Each student is asked to rank all other students in order of preference. All students prefer to be paired with anyone than to work alone.

(a) In this context, what is a matching? What makes a matching stable?
(Get inspiration from the definitions from the Stable Matching problem in class)

Solution:

(b) In the case of $n = 3$, provide a set of preferences for which no stable matching exists. Prove it.

Solution:

(c) In the case of $n = 4$, provide a set of preferences for which no stable matching exists. Prove it. Hint: build upon your answer to the previous question.

Solution:

(d) Find all stable matchings for the following set of preferences. Prove it.

Students
$s_1 : s_2 \succ s_3 \succ s_4$
$s_2 : s_3 \succ s_4 \succ s_1$
$s_3 : s_4 \succ s_1 \succ s_2$
$s_4 : s_1 \succ s_2 \succ s_3$

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 $2 \cdot \log_2 n + 1$.

Solution: