CS3000: Algorithms & Data — Summer I '21 — Drew van der Poel

Homework 1
Due Friday, May 21 at 11:59pm 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, May 21 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. *Proof by Induction (8 points)*

(a) **[8 points]** 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:

Inductive Hypothesis: Let $H(k)$ be the statement: $\sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}$. We will prove that $H(k)$ is true for every $k \in \mathbb{N}$.

Base Case: Consider $H(1)$. $\sum_{i=1}^{1} i^2 = 1$ and $\frac{1(1+1)(2\cdot 1+1)}{6} = 1$. Therefore $H(1)$ holds.

Inductive Step: We will show that for $k \geq 1$, $H(k) \implies H(k+1)$.

$$\begin{aligned}
\sum_{i=1}^{k+1} i^2 &= \sum_{i=1}^{k} i^2 + (k+1)^2 \\
&= \frac{k(k+1)(2k+1)}{6} + (k+1)^2 && \text{(Inductive Hypothesis)} \\
&= (k+1)\left(\frac{k(2k+1)}{6} + k + 1\right) \\
&= (k+1)\left(\frac{2k^2+7k+6}{6}\right) \\
&= \frac{(k+1)((k+1)+1)(2(k+1)+1)}{6}
\end{aligned}$$

Therefore, the claim holds for all n by induction.

Problem 2. *Mystery Code (11 points)*

You encounter the following mysterious piece of code.

Algorithm 1: Mystery Function

```
Function F(a, n):
    If n = 0 :
        Return (1, a)
    Else
        b = 1
        For i from 1 to 2n
            b = b · a
        (u, v) ← F(a, n − 1)
        Return (u · b/a, v · b · a)
```

(a) **[3 points]** What are the results of $F(a,3)$, $F(a,4)$, and $F(a,5)$. You do not need to justify your answers.

Solution:

$$\begin{aligned} F(a,3) &= (a^9, a^{16}) \\ F(a,4) &= (a^{16}, a^{25}) \\ F(a,5) &= (a^{25}, a^{36}) \end{aligned}$$

(b) **[8 points]** What does the code do in general, when given input integer $n \geq 0$? Prove your assertion by induction on n.

Solution: Based on part (a), we guess that the code returns $F(a,n) = (a^{n^2}, a^{(n+1)^2})$ for all integers $n \geq 0$

Inductive Hypothesis: Let $H(n)$ be the statement: $F(a,n) = (a^{n^2}, a^{(n+1)^2})$. We will prove that $H(n)$ is true for every integer $n \geq 0$.

Base Case: By the first branch of the `if-statement`: $F(a,0) = (1,a) = (a^{0^2}, a^{1^2})$. Thus, $H(0)$ is true.

Inductive Step: We will show that for every $n \geq 1$, $H(n-1) \Longrightarrow H(n)$. Assume $H(n-1)$ holds, that is, $F(a,n-1) = (a^{(n-1)^2}, a^{n^2})$. For $n \geq 1$, b is computed by the for loop to be a^{2n}. Then, we have

$$\begin{aligned} (u,v) &= F(a,n-1) \\ &= \left(a^{(n-1)^2}, a^{n^2}\right). && \text{(Inductive hypothesis)} \end{aligned}$$

Thus,

$$\begin{aligned} F(a,n) &= \left(a^{(n-1)^2} a^{2n-1}, a^{n^2} a^{2n+1}\right) \\ &= \left(a^{n^2}, a^{(n+1)^2}\right), \end{aligned}$$

thus establishing $H(n)$. Therefore, $H(n)$ is true for all integers $n \geq 0$ by induction.

Problem 3. *Stable Matching (14 points)*

(a) **[6 points]** State the matching you obtain from running the Gale-Shapley algorithm on the following instance:

hospital	1	2	3
h_1	d_2	d_1	d_3
h_2	d_2	d_3	d_1
h_3	d_1	d_3	d_2

doctor	1	2	3
d_1	h_2	h_3	h_1
d_2	h_3	h_1	h_2
d_3	h_1	h_3	h_2

Is the stable matching you found the only stable matching? If not, provide an example of another stable matching.

Solution:

$(h_1, d_2), (h_2, d_3), (h_3, d_1)$

The above is not the only stable matching. Consider the following: $(d_1, h_2), (d_2, h_3), (d_3, h_1)$ - here all doctors have their top picks, so this is stable.

(b) **[8 points]** Given a set of preferences for n doctors and n hospitals, consider the stable matchings found via the following processes:

(a) Run the standard Gale-Shapley algorithm with hospitals making offers to doctors. Let this matching be M_1.

(b) Run Gale-Shapley again, but this time flip the roles of the hospitals and doctors in the algorithm, so that the doctors make offers to the hospitals. Let this matching be M_2.

Prove the following claim:

If there is more than one stable matching, then $M_1 \neq M_2$.

To do this, you may use the following terminology and Lemma 1.7 from the text. Hospital h is a *valid partner* of doctor d if there is a stable matching the contains the pair (h, d) (and vice versa). Doctor d is the *best/worst valid partner* of h if every other valid partner is ranked lower/higher than d in h's preferences. When hospitals (doctors) propose in Gale-Shapley, each hospital (doctor) is paired with their best valid partner (Lemma 1.7).

Solution:

We will prove this via contradiction.

Assume false, that there is more than one stable matching and $M_1 = M_2$. Let $M*$ be a matching that is different from M_1.

We know there is a doctor d' who is paired with different hospitals in M_1 and $M*$. Let these hospitals be h' and h'' respectively. Now, if d' prefers h'' to h', then h' is not d''s best valid partner. However, this contradicts Lemma 1.7 in M_2.

Since d' must then prefer h' to h'', this means in $M*$, h' must be paired with a doctor d'' that it prefers to d' (otherwise we would have an instability). However, this contradicts Lemma 1.7 in M_1 (as d' cannot be h''s best valid partner).

Problem 4. *Asymptotic Order of Growth (18 points)*

(a) **[10 points]** Rank the following functions in increasing order of asymptotic growth rate. That is, find an ordering $f_1, f_2, \ldots, f_{10}$ of the functions so that $f_i = O(f_{i+1})$. No justification is required.

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

Solution:

$$\begin{aligned}
f_1(n) &= \log_2^3 n \\
f_2(n) &= \sqrt{n} \\
f_3(n) &= 2^{\log_3 n} \\
f_4(n) &= 10n \\
f_5(n) &= \log_2(n!) \\
f_6(n) &= 100n^{3/2} \\
f_7(n) &= n^3 \\
f_8(n) &= 12^n \\
f_9(n) &= 2^{4n} \\
f_{10}(n) &= n!
\end{aligned}$$

(b) **[8 points]** Suppose $f(n), g(n), h(n)$ are non-decreasing, non-negative functions. Decide whether you think the following statement is true or false and give a proof or a counterexample.

If $f(n) = \Omega(h(n))$ and $g(n) = O(h(n))$, then $f(n) = \Omega(g(n))$.

Solution:

This is true.

Because $f(n) = \Omega(h(n))$, there exist constants C' and n_0' such that $f(n) \geq C' * h(n)\ \forall\ n \geq n_0'$.

Because $g(n) = O(h(n))$, there exist constants C'' and n_0'' such that $g(n) \leq C'' * h(n)\ \forall\ n \geq n_0''$.

Note that $C''f(n) \geq C'C''h(n) \geq C'g(n)\ \forall\ n \geq max(n_0', n_0'')$.

Thus, we can let $C = C'/C''$ and $n_0 = max(n_0', n_0'')$, and we see that $f(n) \geq Cg(n)\ \forall\ n \geq n_0$ holds.

Thus, $f(n) = \Omega(g(n))$.