UNIVERSITY OF TORONTO
Faculty of Arts and Science

Midterm 2, Version 1
CSC165H1S

Date: Thursday March 23, 6:10-7:00pm
Duration: 50 minutes
Instructor(s): David Liu, Toniann Pitassi

No Aids Allowed

Name:

Student Number:

Please read the following guidelines carefully!

- **Please write your name on both the front and back of this exam**.
- This examination has **4** questions. There are a total of **8 pages, DOUBLE-SIDED**.
- Answer questions clearly and completely. Provide justification unless explicitly asked not to.
- Formal proofs should follow the same guidelines from the first half of the course (e.g., explicitly introduce all variables and assumptions, clearly state all assumptions and reasoning you make in your proof body, etc.)
- For algorithm analysis questions (including worst-case and best-case), you may freely use external properties of Big-Oh/Omega/Theta presented in the course. You can jump immediately from a step count to an asymptotic bound without proof (e.g., say "the number of steps is $3n + \log n$, which is $\Theta(n)$").
- For all other questions, you may *not* use these properties, or other external facts about definitions introduced in this course, unless explictly allowed to.

Take a deep breath.

This is your chance to show us
How much you've learned.
We **WANT** to give you the credit
That you've earned.
A number does not define you.

Good luck!

Use this page for rough work. If you want work on this page to be marked, please indicate this clearly *at the location of the original question.*

1. **[5 marks] Induction.** Prove the following statement using induction on n:

$$\forall m \in \mathbb{Z}^+,\ \forall n \in \mathbb{N},\ n \geq 2 \Rightarrow m^n + 3 \leq (m+1)^n$$

Hint: $(m+1)^{n+1} = (m+1)(m+1)^n$.

Solution

Proof. Let $m \in \mathbb{Z}^+$, and let $P(n)$ be the predicate "$m^n + 3 \leq (m+1)^n$." We prove that for all $n \in \mathbb{N},\ n \geq 2 \Rightarrow P(n)$.

Base case: let $n = 2$. We want to prove that $m^2 + 3 \leq (m+1)^2$.
In this case, we can calculate:

$$\begin{aligned}
(m+1)^2 &= m^2 + 2m + 1 \\
&\geq m^2 + 2 + 1 && \text{(since } m \geq 1\text{)} \\
&= m^2 + 3
\end{aligned}$$

Induction step: let $n \in \mathbb{N}$, and assume that $n \geq 2$ and that $m^n + 3 \leq (m+1)^n$. We want to prove that $m^{n+1} + 3 \leq (m+1)^{n+1}$. We can calculate:

$$\begin{aligned}
(m+1)^{n+1} &= (m+1)(m+1)^n \\
&\geq (m+1)(m^n + 3) && \text{(by the I.H.)} \\
&= m^{n+1} + m^n + 3m + 3 \\
&\geq m^{n+1} + 3 && \text{(since } m \geq 0\text{)}
\end{aligned}$$

□

2. **[6 marks] Worst-case runtime.** Consider the following algorithm, which takes as input a list of integers.

```
def alg(A):
    n = len(A)
    count = 0
    for i in range(n):          # Loop 1
        if A[i] >= count:
            count = count + 1

    for j in range(count):      # Loop 2
        for k in range(j):      # Loop 3
            print('Counted!')
```

Let $WC(n)$ be the worst-case runtime function of `alg`, where n is the length of the input list A. You can use the following formula in your analysis of $WC(n)$:

$$\forall m \in \mathbb{N}, \ \sum_{i=1}^{m} i = \frac{m(m+1)}{2}$$

Note: assume the integers stored in A can be arbitrarily large (i.e., don't assume some upper limit on the numbers in A).

(a) **[4 marks]** Find, with proof, a good asymptotic upper bound (Big-Oh) on $WC(n)$. By "good" we mean that if you prove $WC \in \mathcal{O}(f)$ (where you chose the f), it should be true that $WC \in \Omega(f)$ as well (but don't prove this here).

Solution

Let $n \in \mathbb{N}$, and consider the running time of `alg` on an input of length n. Loop 1 takes n steps, since there are n iterations, and each iteration takes a single step.

Now let's consider Loops 2 and 3. For a fixed iteration of Loop 2, Loop 3 takes j steps (j iterations, and 1 step per iteration). After Loop 1 ends, *count* has value at most n (since at most it increases by one each iteration). So then Loop 2 takes at most n iterations, for $j = 0, 1, \ldots, n-1$. So then the total cost of Loop 2 is $\displaystyle\sum_{j=0}^{n-1} j = \frac{n(n-1)}{2}$ steps.

Therefore the total runtime is $n + \dfrac{n(n-1)}{2}$ steps, which is $\mathcal{O}(n^2)$.

(b) **[2 marks]** Describe an input family whose runtime matches the upper bound you proved in part (a). For example, if you proved that $WC(n) \in \mathcal{O}(n)$, for this part you should describe an input family whose runtime is $\Theta(n)$.

Only a description of the input family is necessary; you do **not** need to analyse the running time of `alg` on your chosen input family.

Solution

For each $n \in \mathbb{N}$, consider the list A where $A[i] = i$ for each $i \in \{0, \ldots, n-1\}$.

3. **[4 marks] Best-case runtime.** Let $BC(n)$ be the best-case running time of the algorithm `alg` from Question 2. Prove that $BC(n) \in \mathcal{O}(n)$, where n represents the length of the input list. You may assume that $n > 0$ for this analysis.

Solution

We need to find an input family for `alg` whose running time is $\mathcal{O}(n)$. For each $n \in \mathbb{N}$, choose the input list of length n whose elements are all equal to -1.

Consider what happens when we run `alg` on the input list of length n. First, Loop 1 will run n times, with each iteration taking a single step. The cost here is n.

The value of variable *count* is equal to 0 after Loop 1 ends, since the condition `A[i] >= i` is always false.

So then Loops 2 and 3 don't run any times, and so the total cost is n, which is $\mathcal{O}(n)$.

4. **[5 marks] Properties of Big-Oh.** For all functions $f, g \in \mathbb{N} \to \mathbb{R}^{\geq 0}$, we define their *sum function*, denoted $f+g$, to be the following function:

$$(f+g)(n) = f(n) + g(n) \quad \text{for all } n \in \mathbb{N}.$$

Prove that for all functions $f_1, f_2, g_1, g_2 : \mathbb{N} \to \mathbb{R}^{\geq 0}$, if $g_1 \in \mathcal{O}(f_1)$ and $g_2 \in \mathcal{O}(f_2)$, then $g_1 + g_2 \in \mathcal{O}(f_1 + f_2)$.

Reminder: you may not use any properties of Big-Oh in this question. You should use the definition of Big-Oh:

$$g \in \mathcal{O}(f) : \quad \exists c, n_0 \in \mathbb{R}^+,\ \forall n \in \mathbb{N},\ n \geq n_0 \Rightarrow g(n) \leq cf(n), \qquad \text{where } f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$$

<u>Solution</u>

Proof. Let $f_1, f_2, g_1, g_2 : \mathbb{N} \to \mathbb{R}^{\geq 0}$. Assume that $g_1 \in \mathcal{O}(f_1)$ and $g_2 \in \mathcal{O}(f_2)$, i.e., assume there exist $c_1, n_1, c_2, n_2 \in \mathbb{R}^+$ such that for all $n \in \mathbb{N}$:

(i) if $n \geq n_1$, then $g_1(n) \leq c_1 f_1(n)$

(ii) if $n \geq n_2$, then $g_2(n) \leq c_2 f_2(n)$

We want to prove that $g_1 + g_2 \in \mathcal{O}(f_1 + f_2)$. Let $c_3 = c_1 + c_2$ and $n_3 = n_1 + n_2$.* Let $n \in \mathbb{N}$, and assume that $n \geq n_3$. We want to prove that $g_1(n) + g_2(n) \leq c_3(f_1(n) + f_2(n))$.

Since $n \geq n_3 = n_1 + n_2$, we know that $n \geq n_1$ and $n \geq n_2$. So then by assumption (i), we know that $g_1(n) \leq c_1 f_1(n)$, and by assumption (ii) we know that $g_2(n) \leq c_2 f_2(n)$.

Adding these inequalities gives us the inequality

$$\begin{aligned}
g_1(n) + g_2(n) &\leq (c_1 f_1(n)) + (c_2 f_2(n)) \\
&\leq (c_1 + c_2) f_1(n) + (c_1 + c_2) f_2(n) && \text{(since } c_1, c_2 \leq c_1 + c_2\text{)} \\
&= (c_1 + c_2)(f_1(n) + f_2(n)) \\
&= c_3(f_1(n) + f_2(n))
\end{aligned}$$

□

* $c_3 = \max(c_1, c_2)$ and $n_3 = \max(n_1, n_2)$ is also acceptable.

Use this page for rough work. If you want work on this page to be marked, please indicate this clearly *at the location of the original question.*

Name:

Question	Grade	Out of
Q1		5
Q2		6
Q3		4
Q4		5
Total		20