UNIVERSITY OF TORONTO
Faculty of Arts and Science

Midterm 2, Version 2
CSC165H1S

Date: Friday March 24, 12:10-1:00pm
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:

$$\forall n \in \mathbb{N},\ n \geq 1 \Rightarrow 3 \mid 4^{n-1} + 5$$

You may use the following external fact about divisibility, as long as you clearly state where you are using it in your proof.

$$\forall n, a, b, p, q \in \mathbb{Z},\ n \mid a \wedge n \mid b \Rightarrow n \mid ap + bq \tag{Fact 1}$$

Hint: $4^n = 4^n - 4^{n-1} + 4^{n-1}$.

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 count >= n:
            print('Reached limit')
            break
        else:
            count = count + A[i]

    for j in range(min(count, n)):    # Loop 2
        for k in range(j):            # Loop 3
            print('Hello')
```

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) 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).

(b) 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.

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.

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

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

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 $g_1 \times g_2 \in \mathcal{O}(f_1 \times f_2)$.

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

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

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