UNIVERSITY OF TORONTO

Faculty of Arts and Science

term test #2, Version 1
CSC165H1S

Date: Tuesday November 28, 3:10–4:00pm
Duration: 50 minutes
Instructor(s): Danny Heap

No Aids Allowed

Name:

utorid:

U of T email:

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 **7 pages, DOUBLE-SIDED**.
- Answer questions clearly and completely. Provide justification unless explicitly asked not to.
- All formulas must have negations applied directly to propositional variables or predicates.
- In your proofs, you may always use definitions of predicates from the course. You may *not* use any external facts about rates of growth, divisibility, primes, or greatest common divisor unless you prove them, or they are given to you in the question.

Take a deep breath.
This is your chance to show us
How much you've learned.

Good luck!

1. **[5 marks] Induction.** Recall the definition of divisibility:

   $j \mid k$: $\exists i \in \mathbb{Z}, k = ji$, for $j, k \in \mathbb{Z}$

   Now prove the following statement using induction on n:

$$\forall n \in \mathbb{N}, 5 \mid 4^{2n} - 1$$

2. **[5 marks] Properties of Big-Oh.** Recall the following definitions:

- For all functions $f, g \in \mathbb{N} \to \mathbb{R}^{\geq 0}$, we say $f \in \mathcal{O}(g)$ when:

$$\exists c, n_0 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_0 \Rightarrow f(n) \leq cg(n)$$

- For any real number x, $\lceil x \rceil$ is the smallest integer that is no smaller than x, and we may use the following characterization of $\lceil x \rceil$:

$$x \leq \lceil x \rceil < x + 1$$

  Define the function $\lceil f \rceil(n)$ as $\lceil f(n) \rceil$.

- Function f **eventually dominates** 1 if:

$$\exists n_1 \in \mathbb{R}^+, \forall n \in \mathbb{N}, n \geq n_1 \Rightarrow f(n) \geq 1$$

Use these definitions (you may not use any of the properties of big-Oh from the course notes) to prove that if $f \in \mathcal{O}(g)$ and f eventually dominates 1, then $\lceil f \rceil \in \mathcal{O}(g)$. Begin by writing a statement, in predicate logic, of what you aim to prove.

3. **[6 marks] Worst-case runtime**

   Consider the following algorithm:

```
def algorithm(L):
  # assume L is a non-empty list of 0s and 1s
  n = len(L)
  parity = L[0]
  last_switch = 0
  for i in range(n):                    # loop 1
      if parity != L[i]:
          last_switch = i
      parity = L[i]

  for j in range(last_switch):          # loop 2
      for k in range(j):                # loop 3
          print("beep!")
```

   Define $n = \text{len}(L)$ and $WC(n)$ as the worst-case runtime function of `algorithm`. You may find the following formula useful:

$$\sum_{i=0}^{m} i = \frac{m(m+1)}{2}$$

   (a) **[4 marks]** Find, and prove, a tight upper bound on $WC(n)$. By "tight" we mean that if you choose f so that $WC \in \mathcal{O}(f)$ you should be convinced (but no need to prove) that $WC \in \Omega(f)$ also. Begin by writing a statement, in predicate logic, of what you aim to prove.

   (b) **[2 marks]** Describe an input family for `algorithm` whose runtime is in big-Theta of the upper bound from the previous part. Explain your conclusion. No proof is necessary

4. **[3 marks]** Describe an input family whose runtime is in $\mathcal{O}(n)$. Explain your conclusion. No proof is necessary.

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

This page is left nearly blank 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		5
Q3		6
Q4		3
Total		19