UNIVERSITY OF TORONTO

Faculty of Arts and Science

term test #2, Version 1
CSC236F

Date: Thursday November 15, 6:10–7:00pm
Duration: 50 minutes
Instructor(s): Danny Heap

Examination Aids: pencils, pens, erasers, drinks, snacks

---

first and last names:

utorid:

student number:

Please read the following guidelines carefully!

- **Please write your name, utorid, and student number on the front of this exam**.
- This examination has **2** questions. There are a total of **5 pages, DOUBLE-SIDED**.
- Answer questions clearly and completely.
- You will receive 20% of the marks for any question you leave blank or indicate "I cannot answer this question."

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

We **WANT** to give you the credit

Good luck!

1. **[13 marks]** ($\approx$ 35 minutes)

   Define $T(n)$ by:

$$T(n) = \begin{cases} 0 & \text{if } n < 2 \\ n + T(n-2) & \text{if } n \geq 2 \end{cases}$$

   (a) **[3 marks]** Let $q \in \mathbb{N}$ and let $r \in \{0, 1\}$. Use the method of repeated substitution (unwinding) to find a conjecture for a closed form for $T(2q + r)$, that is some function c, using a fixed number of elementary operations, such that $c(2q + r) = T(2q + r)$. You may assume that if $n \in \mathbb{N}^+$, then $\sum_{i=1}^{i=n} i = n(n + 1)/2$, if that assumption turns out to be useful.

   (b) **[6 marks]** Let $r \in \{0, 1\}$, and let function c be your conjecture for a closed form in part (a). Use induction on q to prove $\forall q \in \mathbb{N}, c(2q + r) = T(2q + r)$.

   If you did not find a sucessful conjecture for c in part (a), for up to 4/6 marks you may show that $\forall q \in \mathbb{N}, T(2q + r) \quad q^2$. If you choose this option we will **not** grade any attemptto earn the full 6/6.

(c) **[4 marks]** Prove that $\forall n \in \mathbb{N}^{+}, T(n) - T(n-1) \geq 0$. In other words, prove that T is nondecreasing on $\mathbb{N}$. You may assume, as a consequence of the Quotient/Remainder Theorem, that $\forall n \in \mathbb{N}, \exists q, r \in \mathbb{N}, n = 2q + r \wedge 2 > r$.

2. **[5 marks]** ($\approx$ 15 minutes)

Read over function **reversi** below:

```python
def reversi(a_list: list) -> list:
    """
    Return a copy of a_list reversed.

    Precondition: a_list is a Python list.

    Postcondition: Returns a_list[ : : -1], which is a_list[ : ]
                   in reverse order
    """
    if len(a_list) < 1:
        return a_list[ : ]
    else:
        return reversi(a_list[1 : ]) + [a_list[0]]
```

Use induction on the size of **a list** to prove that **reversi's** precondition, plus execution, implies its postcondition. Assume that the reverse-strideslice [1, 2, 3, 4][: : -1] returns [4, 3, 2, 1], that is the reverse of the original list.