UNIVERSITY OF TORONTO

Faculty of Arts and Science

APRIL 2017 EXAMINATIONS

CSC165H1S

Duration: 3 hours

Instructor(s): David Liu, Toniann Pitassi

No Aids Allowed

Name:

Student Number:

Please read the following guidelines carefully!

- This examination has **9** questions. There are a total of **18 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.
- For algorithm analysis questions, you can jump immediately from a step count to an asymptotic bound without proof (e.g., write "the number of steps is $3n + \log n$, which is $\Theta(n)$").
- You must earn a grade of **at least 40% on this exam to pass this course**.

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.

It's been a real pleasure teaching you this term.

Good luck!

Question	Grade	Out of
Q1		3
Q2		8
Q3		9
Q4		7
Q5		8
Q6		10
Q7		9
Q8		9
Q9		12
Total		75

1. **[3 marks] Propositional logic.** Consider the following propositional formula: $\neg q \Rightarrow (\neg(p \Rightarrow q))$.

   (a) **[1 mark]** Give a truth table for the above formula.

   (b) **[1 mark]** Write the negation of the above formula. You do not need to show any work.

   (c) **[1 mark]** Write a formula that is equivalent to the original formula, and that uses only p, q, and the negation and OR operators. You do not need to show any work.

2. **[8 marks] Predicate logic.** Suppose we have predicates $P(n)$ and $Q(n,m)$ with domains $\mathbb{N}$ and $\mathbb{N} \times \mathbb{N}$, respectively (so Q is a binary predicate that takes two natural numbers).

   (a) [**4 marks**] Translate the following statement into predicate logic:

   "For all natural numbers m and n, $Q(n,m)$ is true if and only if the two numbers are different and the larger of the two numbers satisfies P."

   In your formula, you may only use the standard propositional operators, quantifiers, and the comparison operators $=$, $\neq$, $<$, $\leq$, $>$, and $\geq$. You may *not* use any other symbols or functions; in particular, you may not use the max function.

   (b) [**4 marks**] Consider this pair of statements:

   (1) $\forall x \in \mathbb{N},\ \exists y \in \mathbb{N},\ Q(x,y)$ (2) $\forall x_1, x_2, z \in \mathbb{N},\ x_1 \neq x_2 \Rightarrow \neg Q(x_1, z) \vee \neg Q(x_2, z)$

   Define the predicate Q (with domain $\mathbb{N} \times \mathbb{N}$) so that both of these statements are true, and briefly explain your answer (no formal proofs necessary). You're only defining this predicate once; the two statements must use the same definition for Q.

   Note: this part is completely independent of part (a).

3. [**9 marks**] **Proofs.** Your goal for this question is to prove that the equation $x^2 - 3 = 4y$ has no integer solutions. We'll break this down into parts.

   (a) [**4 marks**] Prove that for all integers x, $4 \mid x^2$ or $4 \mid x^2 - 1$.

(b) **[1 mark]** Translate the following statement into predicate logic: "the equation $x^2 - 3 = 4y$ has no integer solutions for x and y."

(c) **[4 marks]** Prove the statement from part (b). You may use the statement from part (a), the definition of divisibility, and the following fact about divisibility:

$$\forall n, a, b, p, q \in \mathbb{Z},\ n \mid a \land n \mid b \Rightarrow n \mid ap + bq \qquad \text{(Fact 1)}$$

Hint: use a proof by contradiction.

4. [**7 marks**] **Induction.** Prove the following statement by induction on n:

$$\forall a \in \mathbb{Z}^+,\ \forall n \in \mathbb{N},\ n \geq a \Rightarrow \sum_{i=a}^{n} \frac{a}{i(i+1)} = 1 - \frac{a}{n+1}$$

Hint: the variable a should be introduced first in your proof. The base case should be when $n = a$.

5. **[8 marks] Big-Oh/Omega/Theta properties.** For each function $f \in \mathbb{N} \to \mathbb{R}^{\geq 0}$, we define its corresponding *ceiling function*, denoted $\lceil f \rceil$, to be the following function:

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

Prove the following statement:

$$\forall f, g : \mathbb{N} \to \mathbb{R}^{\geq 0},\ g \in \Omega(f) \wedge \big(\forall m \in \mathbb{N},\ f(m) \geq 1\big) \Rightarrow g \in \Omega(\lceil f \rceil)$$

You may not use any properties of Omega in this question; you should use the definition of Omega: $g \in \Omega(f) :\ \exists c, n_0 \in \mathbb{R}^+,\ \forall n \in \mathbb{N},\ n \geq n_0 \Rightarrow g(n) \geq cf(n)$.

You may also use the fact that "for all $x \in \mathbb{R}$, $x \leq \lceil x \rceil < x + 1$" as long as you clearly state when you do.

Hint: $cf(n) = \frac{c}{2}f(n) + \frac{c}{2}f(n)$.

6. **[10 marks] Runtime analysis (1).** Each of the following functions takes as input a positive integer n, and has a running time that depends only on n.

(a) **[4 marks]**

```
1 def f1(n):
2     i = 0
3     while i < n * n:          # Loop 1
4         j = 1
5         while j < n:          # Loop 2
6             j = j * 4
7         i = i + 2
```

For simplicity, we will consider the runtime of `f1` to be only the cost of executing Line 6 (ignoring all other operations).

Determine the exact number of times Line 6 is executed, in terms of the function's input n.

(b) **[1 mark]** Using your answer to part (a), determine a simple Theta expression for the runtime of `f1`. No justification is required.

(c) **[4 marks]** Assume that we have a helper function `h(m)` whose running time is exactly m^2 steps, where m is the value of its input. For example, the call `h(10)` takes exactly 100 steps.

For simplicity, we will consider the runtime of `f2` to be only the cost of calling the helper function `h` at Line 3 (ignoring all other operations). Determine the exact total cost of all executions of Line 3 of the algorithm below.

```
1 def f2(n):
2     for j in range(n*n):
3         h(j)
4         print(j)
```

You can use the following formula, which is valid for all $k \in \mathbb{N}$:

$$\sum_{i=0}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}$$

(d) **[1 mark]** Using your answer to part (c), determine a simple Theta expression for the runtime of `f2`. No justification is required.

7. **[9 marks] Runtime analysis (2).** The following function takes as input a list A that is a permutation of $\{0, 1, 2, \ldots, len(A) - 1\}$. For example, the lists `[0, 2, 1]` and `[5, 4, 3, 2, 1, 0]` are valid inputs.

```
def alg(A):
    n = len(A)
    i = 0
    for j in range(n):
        if A[i] == 0:
            return i
        else:
            i = A[i]

    return -1
```

Let $WC(n)$ be the worst-case runtime function of `alg`, where n is the length of the list A.

(a) **[3 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 choose the f), it should be possible to prove a matching lower bound, $WC \in \Omega(f)$, in part (b).

(b) [**3 marks**] Prove a lower bound (Omega) on $WC(n)$ that matches the upper bound you proved in part (a). For example, if you proved in part (a) that $WC(n) \in \mathcal{O}(n^2)$, for this part you should prove that $WC(n) \in \Omega(n^2)$.

(c) [**3 marks**] Find, with proof, a good asymptotic upper bound (Big-Oh) on the best-case runtime $BC(n)$ of `alg`.

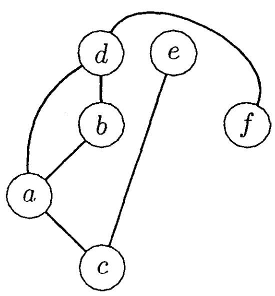
8. [**9 marks**] **Graphs.** This question is an exercise in reading definitions and applying them to a new situation. Please read the following definitions carefully, and then answer the questions below.

Definition 1 (vertex removal). Let $G = (V, E)$ be a graph, and let $v \in V$ be a vertex in G. We define the graph $G - v$ to be the graph $G' = (V', E')$, where:

- $V' = V \backslash \{v\}$. (the vertices of G except for v)
- $E' = \{(u, w) \mid (u, w) \in E, \text{ where } u \neq v \wedge w \neq v\}$. (the edges of G that do not touch v).

Definition 2 (cut vertex). Let $G = (V, E)$ be a *connected* graph, and let $v \in V$ be a vertex in G. We say that v is a **cut vertex** of G when the graph $G - v$ is not connected.

(a) [**2 marks**] Below is a graph $G_1 = (V_1, E_1)$, where $V_1 = \{a, b, c, d, e, f\}$. Beside it, draw the graph $G_1 - a$. Include all vertex labels in your drawing.



G_1

Draw $G_1 - a$ here.

(b) [**2 marks**] You should find that the graph $G_1 - a$ is not connected; this means that vertex a is a cut vertex of G_1. List all of the other cut vertices of the original graph G_1. No justification is required.

(c) **[5 marks]** Recall that a tree is a graph that is connected and contains no cycles. Prove that for all trees $G = (V, E)$ and all vertices $v \in V$, if v has at least two neighbours, then v is a cut vertex of G.

9. **[12 marks] Average-case analysis.** Recall that a **binary string** is a string where every character is 0 or 1. Consider the following algorithm, which takes as input a binary string.

   NOTE: to simplify the calculations, we'll assume the input A is indexed starting at 1, not 0 (so the elements are $A[1], A[2], \ldots, A[len(A)]$)). Furthermore, we'll assume that the length of A is a power of two.

```
def newalg(A):
    n = len(A)
    i = 1
    while i <= n:
        if A[i] == 0:
            return i
        else:
            i = 2 * i
    return -1
```

Our goal is to perform an average-case analysis on this algorithm. For this question, the **set of allowed inputs** are all possible binary strings of length n, where the last character is a 0. For example, if $n = 4$, then the set of all inputs is $\{0000, 0010, 0100, 0110, 1000, 1010, 1100, 1110\}$.

(a) **[2 marks]** Give an exact formula for the number of allowed inputs of length n. No justification is required.

(b) **[2 marks]** Let $b \in \mathbb{N}$ and $n = 2^b$. What is the minimum number of iterations of the while loop on inputs of length n? Give a description of the set of the allowed inputs of length n that achieve this minimum, and state the size of this set.

(c) **[3 marks]** Let $b \in \mathbb{N}$ and $n = 2^b$. What is the maximum number of iterations of the while loop on inputs of length n? Give a description of the set of the allowed inputs of length n that achieve this maximum, and state the size of this set.

(d) **[4 marks]** Let $b \in \mathbb{N}$ and $n = 2^b$. Determine an exact expression for the average-case runtime of the above algorithm (over the set of all inputs of length n). This expression can contain a summation.

(e) **[1 mark]** Finally, use the expression from part (d) to get a good Big-Oh bound on the average-case runtime of this algorithm. You may use the fact that for all $x \in \mathbb{R}$ such that $|x| < 1$, $\sum_{i=1}^{\infty} ix^i = \frac{x}{(1-x)^2}$.

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.*

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.*

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.*