University of Waterloo
CS240 - Spring 2017
Assignment 1

Due Date: Wednesday, May 17th, 5pm

Please refer to the course webpage for guidelines on submission. Submit your written solutions electronically as a PDF with file name `a01wp.pdf` using MarkUs. We will also accept individual question files named `a01q1w.pdf`, `a01q2w.pdf`, etc., if you wish to submit questions as you complete them.

Unless the base is explicitly specified, we assume in all questions that the logarithm operation is base 2.

Problem 1 [4+4+4+4=16 marks]

Provide a complete proof of the following statements from first principles (i.e., using the original definitions of order notation). You may use any mathematical facts presented in lecture or in the course notes without proof.

(a) $42n^8 + 16n^6 + 5n\log(n) + 2017 \in O(n^{10})$.

(b) $n^3(\log(n))^2 \in \Omega(n^3)$.

(c) $n^5/(n^3 + 3n) \in \Theta(n^2)$.

(d) $n^n \in \omega(n^{100})$.

Problem 2 [4+4+4=12 marks]

Prove or disprove each of the following statements. To prove or disprove a statement, you must provide a proof based on the definitions of each order notation. However, it is sufficient to provide an example with justification to prove an existence statement or to disprove a universal statement.

(a) There exist functions $f(n)$ and $g(n)$ such that $f(n) \in o(g(n))$ and $f(n) \in \omega(g(n))$.

(b) For all $k \geq 1$, $(\log(n))^k \in o(n)$.

(c) $1 + n + n^2 + \cdots + n^{k-1} \in \Theta(n^k)$.

Problem 3 [4+4+4=12 marks]

Give an analysis of the running times of each of the following blocks of pseudocode using order notation. A formal proof is not required, but you should show your work and justify your answer. For full marks, your bounds must be as tight as possible.

(a)
```
for i from 1 to n*n do
    j = n
    while j > 0 do
        j = j - 1
```

(b)
```
x = 0
for i from 1 to n*n do
    for j from 1 to i*i do
        x = x + 1
```

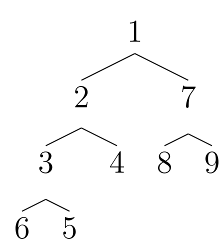
(c)
```
x = 1
for i from 1 to n do
    x = x * 2
    for j from 1 to x do
        y = y + 1
```

Problem 4 [5+5=10 marks]

(a) Heapify the following array in a min-heap using the techniques presented in the course notes. List each heap operation being performed and on which element, and present the state of the heap after each operation.

4	2	7	5	1	8	9	6	3

(b) Remove the keys 1, 2, 3, 4, and 5 from the following min-heap in that order. List each comparison and operation being performed and on which element, and present the state of the heap after each operation.



Problem 5 [2+3(+5 bonus)=5(+5) marks]

A *ternary heap* is a generalization of heaps that uses a ternary tree, or a tree where each node may have up to three children. Ternary heaps share all of the same properties as regular heaps regarding the insertion and position of nodes; namely, all levels except possibly the bottom level of a ternary heap are filled, the bottom level of a ternary heap is filled from left to right, and the value stored in a node must be larger than the values stored in any of the node's children.

(a) Explain how you can store a ternary heap as an array A of size $O(n)$ such that the root of the ternary heap is located at position $A[0]$. Explain how you can find the parent and all children of an arbitrary node $A[i]$ in the ternary heap.

(b) Prove that the height of a ternary heap with n nodes is at most $\frac{\log(2n-1)}{\log(3)}$.

(c) *Bonus.* Given two ternary heaps H_1 and H_2 that are both of the same height and both full, describe an algorithm to *merge* H_1 and H_2; that is, to create a new ternary heap H that contains all elements in both H_1 and H_2. The running time of your algorithm should be $O(\log(n))$, where $n = |H_1| = |H_2|$. You do not need to prove the correctness of your algorithm, but you must justify the running time.

You may assume that H_1, H_2, and H are implemented as tree data structures rather than arrays, and that all elements in H_1 and H_2 are distinct.