YORK UNIVERSITY
Faculty of Pure & Applied Science

Dept. of Computer Science
October 22, 1997

MID-TERM TEST
COSC3101.03: Design & Analysis of Algorithms

- Do all problems.
- Put all answers in this booklet.
- Exam period is 1.5 lecture-hours.
- Closed book.
- Do *not* hand in anything other than this booklet.
- You may use back side of pages for scratch work.

Name:____________________________________

Student Number:___________________________

Problem	Points Received	Points Worth
1		20
2		25
3		30
4		25
TOTAL		100

Problem 1. [20%]

For each of the 10 statements that follow indicate only whether it is true or false by circling T (true) or F (false). Do *not* justify your answer. Each correct answer is worth +2 points. Each incorrect answer or no answer is worth 0 point.

(a) T F
If $f(n) = O(g(n))$, and both $f(n)$ and $g(n)$ are ≥ 2, then $lg\ (f(n)) = O(\ lg\ (g(n))\ )$.

(b) T F
If $f(n) = O(g(n))$, and both $f(n)$ and $g(n)$ are ≥ 2, then $2^{f(n)} = O(\ 2^{g(n)}\ )$.

(c) T F
$\Theta(n^2) \cdot O(n^3\ lg\ n) = O(\ n^6\ /\ lg\ n\ )$.

(d) T F
$(n)^{lg\ n} + (lg\ n)^n + 2^{n\ lg\ n} = \Theta(n^n)$.

(e) T F
$\sum_{i=1}^{n} \frac{i}{2^i} = \Theta(\ \sum_{i=1}^{n} \frac{1}{2^i}\ )$.

(f) T F
The solution to the recurrence $T(n) = T(\frac{n}{5}) + T(\frac{4n}{5}) + n$ is $T(n) = \Theta(n\ lg\ n)$.

(g) T F
Worst case time complexity of QuickSort is $\Theta(n\ lg\ n)$.

(h) T F
Average case time complexity of QuickSelect is $\Theta(n)$.

(i) T F
For all n, the smallest possible depth of a leaf in a decision tree for sorting n elements is 1.

(j) T F
To find the closest pair among n points on the 2 dimensional plane requires at least $\Omega(n^2)$ time in the worst case.

Problem 2. **[25%]**

Consider the recurrence relation

$$T(n) = \begin{cases} 8T(\frac{n}{4}) + f(n) & \textit{for } n > 1 \\ \Theta(1) & \textit{for } n \le 1 \end{cases}$$

Derive a tight asymptotic bound on $T(n)$, when

(a) [5%] $f(n) = \sqrt{n}$.

(b) [5%] $f(n) = n\sqrt{n}$.

(c) [15%] $f(n) = \frac{n\sqrt{n}}{lg\ n}$.

Note: you should show your work and mention the methods used.

Problem 3. [30%]

This problem concerns max-heaps.

(a) [5%] Using the tree model, illustrate the operation of *Heap-Extract-Max(A)* on the max-heap $A = [15, 13, 9, 5, 12, 8, 1, 4, 0, 6, 2, 7]$.

(b) [25%] Design an efficient implementation of the procedure *Heap-Increase-Key(A,i,k)*, which sets $A[i] \leftarrow max\ (A[i]\ ,\ k)$ and then updates the heap structure by rearranging its items to reestablish the heap property.

- You should give your algorithm in **detailed and precise pseudo-code**.
- What is the worst case time complexity of your algorithm?

Problem 4. [25%]

We are given 10 distinct numbers $a_1, a_2, \cdots, a_5$ and $b_1, b_2, \cdots, b_5$. We already know that $a_1 < a_2 < a_3 < a_4 < a_5$, and $b_1 < b_2 < b_3 < b_4 < b_5$. Our aim is to find the median of these 10 numbers (i.e., the 5-th smallest), using as few key comparisons in the worst-case as possible.

- Draw a decision tree for one such method.
- What is the worst-case number of comparisons made by your decision tree?