COSC3101 Design and Analysis of Algorithms
Jeff Edmonds & Andy Mirzaian – Fall 00-01
Midterm Test

Family Name:

Given Name:

Student #:

email:

Section:

1 True/False	20 points	
2 Asymptotics	16 points	
3 Sums & Recurrences	21 points	
4 Loop Invariants	20 points	
5 Recursion	23 points	
TOTAL	100 points	

This test is closed book and lasts 90 minutes.
Do not use any electronic/mechanical computation devices.
This booklet contains 5 pages including this cover page.
Read all questions before deciding in what order to answer them.

1. **True/False:** (20 points)
For each of the 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 +4 points. Each incorrect answer or no answer is worth 0 points.

   (a) [**T F**]: $2^{6 \log n} + 23n^7 (\log n)^4 = \mathcal{O}(\frac{n^8}{(\log n)^8})$.

   (b) [**T F**]: The running time of Insertion-Sort is $\Theta(n + I)$, where I is the number of inversions in the input array $A[1..n]$. (An *inversion* is any pair of items $A[i]$ and $A[j]$ such that $A[i] < A[j]$ but $i > j$.)

   (c) [**T F**]: In the worst case QuickSort takes $\Theta(n \log n)$ time to sort n elements.

   (d) [**T F**]: Let $A[1..n]$ be an array of n elements such that we already know $A[i] < A[i+4]$ for all $i = 1, 2, 3, ..., n-4$. Even with this extra information as precondition, every decision tree that completes the sorting of $A[1..n]$ must have height at least $\Omega(n \log n)$.

   (e) [**T F**]: Given an arbitrary sorted array $A[1..n]$ of reals, we can determine whether A has a majority element or not in $\Theta(\log n)$ time in the worst case. (A *majority* element in A is one that appears more than $n/2$ times.)

2. $f(n) = n^{\Theta(1)}$

   (a) (1 point) Informally, which functions are included in the classification $f(n) = n^{\Theta(1)}$?

   (b) (3 points) The formal definition of $f(n) = n^{\Theta(1)}$ includes three parameters c_1, c_2, and n_0. Give this formal definition.

   (c) (12 points) Which of the following are $f(n) = n^{\Theta(1)}$? If so, give suitable values of c_1 and c_2 for when $n_0 = 1000000$.

1) $f(n) = 3n^3 + 17n^2 + 4$	**Yes,** $c_1 =$	$c_2 =$	**No: Why?**
2) $f(n) = 3n^3 \log n$	**Yes,** $c_1 =$	$c_2 =$	**No: Why?**
3) $f(n) = n^{3 \log n}$	**Yes,** $c_1 =$	$c_2 =$	**No: Why?**
4) $f(n) = 3 \log n$	**Yes,** $c_1 =$	$c_2 =$	**No: Why?**
5) $f(n) = 7^{3 \log n}$	**Yes,** $c_1 =$	$c_2 =$	**No: Why?**
6) $f(n) = \lceil \log n \rceil !$	**Yes,** $c_1 =$	$c_2 =$	**No: Why?**

3. **Sums & Recurrences:** (21 points)
Derive tight asymptotic bound solutions to the following. Mention the method you use for each. (For the recurrences you may assume the usual boundary condition: $T(\mathcal{O}(1)) = \mathcal{O}(1)$.)

(a) $\sum_{i=1}^{n} i^8 \times (\log i)^8 = \Theta($

(b) $\sum_{i=1}^{n} 3^{2i} \times i^8 = \Theta($

(c) $\sum_{i=1}^{n} \frac{1}{i^{1.1}} = \Theta($

(d) $T(n) = 2T(n-1) + 1,\ T(n) = \Theta($

(e) $T(n) = 3T(\frac{n}{3}) + 3(\log n)^3,\ T(n) = \Theta($

(f) $T(n) = 9T(n/3) + 7n \log n + 2n^2,\ T(n) = \Theta($

(g) $T(n) = T(n-1) + n,\ T(n) = \Theta($

4. **Iteration & Loop Invariants:** (20 points)
We are given an arbitrary **sorted** array A[1..n] of n real numbers. Some items may appear several times in A. The problem is to find an item that occurs most often in A.

Use iteration and loop invariants to design, describe, and prove the correctness of an incremental algorithm for the above problem. Be sure to include ALL required steps.

5. **Recursion on Binary Trees:** (23 points)
We are given an arbitrary **binary search tree** T. Each node of T stores an item which is a real number. The problem is to find a closest pair of items in T, that is, a pair of items (distinct nodes) in T with minimum possible difference in value.
Design a recursive algorithm for this problem, and prove its correctness by induction.