Name:________________________________

ECE 608: Computational Models and Methods, Fall 2005
Test #1
Monday, October 3, 2005

- Your exam should have 10 (ten) pages.
- Page 9 is intentionally left blank.
- Page 10 contains a list of potentially useful identities that you may use.
- Write your name on this page and at least one other page.
- Closed book, closed notes.
- Switch off and put away cell-phones/pagers
- Apportion your time carefully.
- Numbers in brackets represent points for that question. Points add up to 100.
- Good luck.

Prob.	Max.	Score
I	15	
II	10	
III	10	
IV	15	
V	30	
VI	20	
Total	100	

Name:______________________________

I. (15 points) Categorize the following statements as **True** or **False**.

a. The summation $\sum_{k=10}^{n} \frac{1}{\lg(k)}$ is lower-bounded by the following integral: $\int_{9}^{n} \frac{1}{\lg(x)} dx$ for all $n \geq 15$.

b. The worst-case complexity of BUILDHEAP is $O(n \lg n)$.

c. If $T(n) = \sum_{k=1}^{n} k^i$, for some constant positive integer *i,* then $T(n) = \Theta(n^{i+1})$

d. There may exist a comparison sort algorithm whose worst-case asymptotic time complexity is given by the following recurrence: $T(n) = T\left(\frac{73n}{134}\right) + n^{\frac{1}{\ln(n)}}, n > 1$ and $T(n) = 1, n = 1$.

e. HEAPSORT and QUICKSORT have identical asymptotic time complexity in the worst case.

II. (10 points) Prove using induction that if $af\left(\frac{n}{b}\right) \leq cf(n)$ is true when $a \geq 1, b > 1, n > b$ and f(n) is an asymptotically positive function of n, then $a^i f\left(\frac{n}{b^i}\right) \leq c^i f(n)$ also holds true for all $1 \leq i \leq \log_b n$. Assume that n is an exact power of b.

Name:________________________________

III. (10 points) Summations and Recurrences.

a. (7 points) If $f(n) = \sum_{k=1}^{n} \frac{k}{k+c}$ where $c > 0$ is a constant, prove that $f(n) = \Theta(n)$.

b. (3 points) Is the Master method suitable/applicable to derive the asymptotic complexity of the following recurrence? If so, state the asymptotic complexity. If not, state why the method is not applicable.

$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + 3n^2, n > 1 \\ 1, n = 1 \end{cases}$$

IV. (15 points) Arrange the following functions in increasing order of asymptotic complexity. If multiple functions are of equivalent complexity **OR** if certain functions cannot be compared with others in the list, your answer must state it explicitly.

a. $T(n) = 10T\left(\frac{n}{3}\right) + \Theta(n^2)$

b. The worst-case asymptotic complexity of INSERT_SORT with n-element input

c. $\sin^2(n) + \sqrt{2}$

d. $\lg^{\frac{1}{\lg \lg n}} n$

e. The worst-case asymptotic complexity of BUILDHEAP with n-element input.

V. (30 points) A **k-ary** tree is a tree where each vertex may have upto **k** child-vertices which are numbered from 1 to k. By convention, the child vertices are arranged from left to right in increasing order of numbering. (Note, a binary-tree is a **k-ary** tree with **k=2** and the first child labeled as the left child and the second child labeled as the right child**.**) We define a **k-ary** MAX-HEAP as a **k-ary** tree that satisfies the following two properties: (i) Nearly-complete property: All non-leaf vertices, except for possibly one, have **k** children. If a non-leaf vertex has fewer than **k** children (which must all be left-most children), it is the right-most vertex at that depth with children. Fig 1 illustrates a nearly-complete 3-ary tree. (ii) **Max-Heap property:** The key of the parent must be greater than or equal to the key of the child.

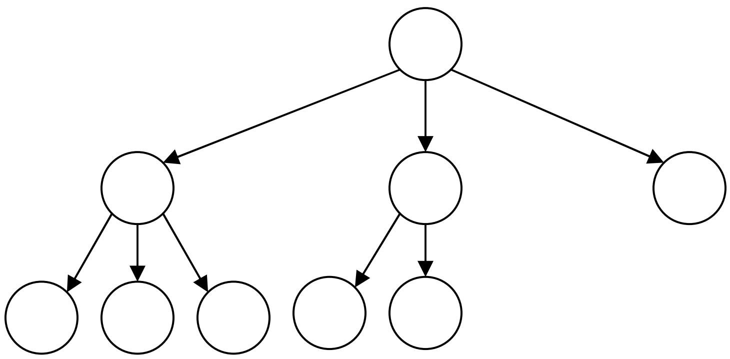

Fig 1: Nearly-complete 3-ary tree

The k-ary heap is to be stored in an array. Provide the pseudocode for the following k-ary heap manipulation primitives. (Hint: Since the traditional binary heap is a special case of the k-ary heap, you can verify the correctness of your pseudocode by testing it with k=2.)

a. (5 points) JTH-CHILD(**i,j,k**) : Returns the index of the $\mathbf{j}^{\mathbf{th}}$ (1≤**j**≤**k**) child of vertex **i** in a **k**-ary heap. Hint: It may be easier to derive the relationship between a parent and its k^{th} (i.e., right-most) child. The index of other child nodes may be calculated using an offset from the k^{th} child. Your code need not include error handling to check for invalid arguments.

b. (5 points) PARENT(i,k) : Returns index of the parent of vertex **i** in a **k**-ary heap.

c. (10 points) KARY-MAX-HEAPIFY(**A, i, k**) : This function must restore the heap property in a **k**-ary max-heap (contained in array **A**) in which only the **i**th element violates the heap property. You may use the PARENT() and JTH-CHILD() primitives developed in parts (a) and (b) above.

d. (10 points) BUILD-KARY-MAX-HEAP(**A, k**) : This function must build a heap from an unordered list of elements supplied in the array **A**. You may use the PARENT(), JTH-CHILD(), and KARY-MAX-HEAPIFY() primitives developed in parts (a), (b) and (c) above.

Name:______________________________

VI. (20 points) The ECE 608 class has developed a new version of PARTITION called **PARTITION608** with the following properties.
   - When the input array contains at least 1217 (=608*2+1) elements, **PARTITION608** selects the pivot element such that each partition has at least 608 elements.
   - When the input array is smaller than 1216, **PARTITION608** returns the index of median of the array as the pivot resulting in an even split.
   - **PARTITION608** runs in linear time.

QUICKSORT608, a minor modification of QUICKSORT, invokes **PARTITION608** instead of the basic PARTITION function. The recursion terminates when QUICKSORT608 is invoked on a single-element array.

a. (5 points) Let T(n) be the asymptotic worst-case time complexity of **QUICKSORT608**. Express T(n) as a recurrence.

b. (15 points) Solve the recurrence obtained in (a) to derive asymptotic tight bounds for the worst-case time complexity of **QUICKSORT608**.

Name:________________________________

Potentially Useful Identities and Approximations

$$\log_2 n = \lg n$$
$$\log_e n = \ln n$$
$$\lg^k n = (\lg n)^k$$
$$\lg \lg n = \lg(\lg(n))$$
$$\log_b a = \frac{1}{\log_a b}$$
$$n^{\log_b a} = a^{\log_b n}$$
$$\log_b a = \frac{\log_c a}{\log_c b}$$

$$\left(a^m\right)^n = \left(a^n\right)^m = a^{mn}$$
$$a^m . a^n = a^{m+n}$$

$$\sum_{k=0}^{n} x^k = \frac{x^{n+1} - 1}{x - 1}$$
$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}, |x| < 1$$
$$\sum_{k=1}^{n} \frac{1}{k} = \ln(n) + \mathrm{O}(1)$$
$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$
$$\sum_{k=1}^{n} k^2 = \frac{n(2n+1)(n+1)}{6}$$
$$\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4}$$

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$$