ECE608 Midterm 1 Spring 2007

Please fill in the following information.

THEN PUT YOUR EMAIL ADDRESS ON EVERY PAGE OF THE EXAM !

NAME: **SOLUTION**

Email (please put complete address):

Neatness counts. We will not grade what we cannot read.

Exam is worth 100 points. Please phrase your answer succinctly. Write your answer on the same page, on which the question is given. There are four questions and a total of nine pages. Make sure you turn all these pages.

Do not attempt to look at other students' work. Keep your answers to yourself.
Any sort of cheating will result in a zero grade.

Read and sign the statement below. Wait for instructions to start the examination before continuing to the next page.

"I signify that the work shown in this examination booklet is my own and that I have not received any assistance from other students nor given any assistance to other students."

_________________________________(Signature)

Q1. Answer the following questions. **(45 points total)**

(a) **(8 points)** Write a recurrence equation that describes the number of lines of output generated by COMPOSE(n). Don't worry about solving the recurrence, but do not forget the base case.

```
COMPOSE (n)
1. for $i \leftarrow 1$ to $n$
2.     do for $j \leftarrow 1$ to $\sqrt{n}$
3.         do write( $i$ , $j$, $n$ )          ▷ print line with $i$, $j$, and $n$
4. if n > 0
5.    then for $i \leftarrow 1$ to 5
6.         COMPOSE ($\lfloor \frac{n}{2} \rfloor$)
```

Solution:

$$T(n) = \begin{cases} 0 & \text{if } n = 0, \\ 5T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n\sqrt{n} & \text{if } n > 0. \end{cases}$$

(b) **(15 points**) Derive the exact closed form for the following summation, assuming that $|a| \geq 1$.

$$\sqrt{\sum_{i=1}^{n} i * a^{i-1}}$$

Solution:

Take the derivative of both sides of the geometric series formula,

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1}$$

We get:

$$\sum_{i=1}^{a} i * a^{i-1} = \frac{(n+1)a^n (a-1) - (a^{n+1} - 1)}{(a-1)^2}$$

After some manipulation we get:

$$= \frac{1}{(1-a)^2}[1 - a^n - na^n + na^{n+1}]$$

$$= \frac{1}{(1-a)^2}[1 - (n+1)a^n + na^{n+1}]$$

The final answer is the square root of this expression.

(c) Solve the following recurrence *T(n)* using Θ-notation. **(15 points total)**

(i) **(5 points)**

$$T(n) = \begin{cases} 1 & if\ n = 1, \\ 1000T(\frac{n}{10}) + \sqrt{n} & if\ n > 1. \end{cases}$$

Solution:

Since $a = 1000$, $b = 10$, $n^{\log_{10} 1000} = n^3$, $f(n) = n^{\frac{1}{2}}$, and $f(n) = 0(n^{2-\in})$ for $1 \le \in \le 1.5$, by case 1 of the Master Theorem, $T(n) = \Theta(n^3)$.

(ii) **(5 points)**

$$T(n) = \begin{cases} 1 & if\ n = 1, \\ 81T\left(\frac{n}{3}\right) + n^4 \lg^2 n & if\ n > 1. \end{cases}$$

Solution:

Since $a = 81$, $b = 3$, $n^{\log_3 81} = n^4$, $f(n) = n^4 \lg^2 n$, and $f(n) = \Theta(n^{\log_3 81} \lg^2 n)$, so by case 2 of the Master Theorem, $T(n) = \Theta(n^4 \lg^3 n)$

(iii). (5 points)

$$T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{4n}{5}\right) + \theta(n)$$

Solution:

Master's theorem doesn't apply here. Draw recursion tree. At each level, do $\Theta(n)$ work. Number of levels is $\log_{5/4} n = \Theta(\lg n)$, so guess $T(n) = \Theta(n \lg n)$ and use the substitution method to verify the guess.

(d) (7 points) A coupon collector tries to collect *m* distinct coupons (each coupon has an equal chance to be collected). After *k* tries, on the average how many times a particular coupon is picked up by the collector?

Solution: *k/m*
(see page 109 of the textbook)

Q2. (20 points total) There are two types of professors who have taught ECE608: ***nice*** professors and ***mean*** professors. The nice professors assign A's to all of their students and the mean professors assign A's to **exactly** 75% of their students and B's to the remaining 25% of the students. For example, for $n = 8$, the arrays [`A A A B A B A A`] and [`A B B A A A A A`] represent grades assigned by mean professors, and the array [`A A A A A A A A`] represents grades assigned by a nice professor. Given an array $G[1 .. n]$ of grades from ECE608, we wish to decide whether the professor who assigned the grades was nice or mean.

a. (15 points) Give an efficient **randomized** algorithm to decide whether a given array G represents grades assigned by a mean or nice professor. Your algorithm should be correct with probability at least 51%, that is, the probability of giving a correct solution is at least 51%.

b. (5 points) What is the running time complexity of your algorithm?

(a)

Here's the algorithm:

```
1. Repeat the following t times
        (a) choose a random index $i \in \{1,...,n\}$
        (b) if $G[i] = B$, then return "mean"
2. Return "nice"
```

First, note that with probability one, a nice professor will be noted as such, since there are no B's in G in this case.
The probability that a mean professor makes it through an iteration of loop is ¾, since ¼ of the entries of array are B's. The probability that the mean professor makes it through t iterations is $(3/4)^t$. If we take t to be 3, then $(3/4)^t < 0.49$. Thus the probability of giving a correct solution is at least 51%.

(b)

The running time is obviously $O(1)$, since we do a constant number of iterations of constant-time loop.

Q3. (10 points) Suppose that the splits at every level of Quicksort are in the proportion 1-α to α, where $0 < \alpha \leq ½$ is a constant. Show that the minimum depth of a leaf in the recursion tree is approximately $-\lg n / \lg \alpha$ and the maximum depth is approximately $-\lg n / \lg(1-\alpha)$. (Don't worry about integer round-off.)

Solution:

The minimum depth follows a path that always takes the smaller part of partition – i.e, that multiplies the number of elements by α. One iteration reduces the number of elements from *n* to αn, and *i* iterations reduces the number of elements to $\alpha^i n$. At a leaf, there is just one remaining element, and so at a minimum depth leaf of depth *m*, we have $\alpha^m n = 1$. Thus, $\alpha^m = 1/n$. Taking logs, we get $m \log \alpha = -\log n$ or $m = -\log n / \log \alpha$.

Similarly, maximum depth corresponds to always taking the larger part of the partition, i.e., keeping a fraction $1-\alpha$ of the elements each time. The maximum depth *M* is reached when there is one element left, that is, when $(1-\alpha)^M n = 1$. Thus, $M = -\lg n / \lg(1-\alpha)$.

All these equations are approximate because we are ignoring floors and ceilings.

Q4. (25 points total)

(a) **(10 points)** One can generalize the definition of binary-heap (the one we studied in the class) to ***d***-heap in which every node has at most *d* children. Describe a labeling scheme for such a heap. In other words, if such a heap is stored in an array, for an entry located at position *i* of the array, find the indices for its parent and children nodes? Assume the first element of the array has index 1.

(b) **(15 points)** Insert the following items in a 4-heap (assume a max heap). You must show your work for full credit.

64, 87, 22, -6, 51, 4, -4, 34, 67, 79, -7, 23, 26, 1, 24

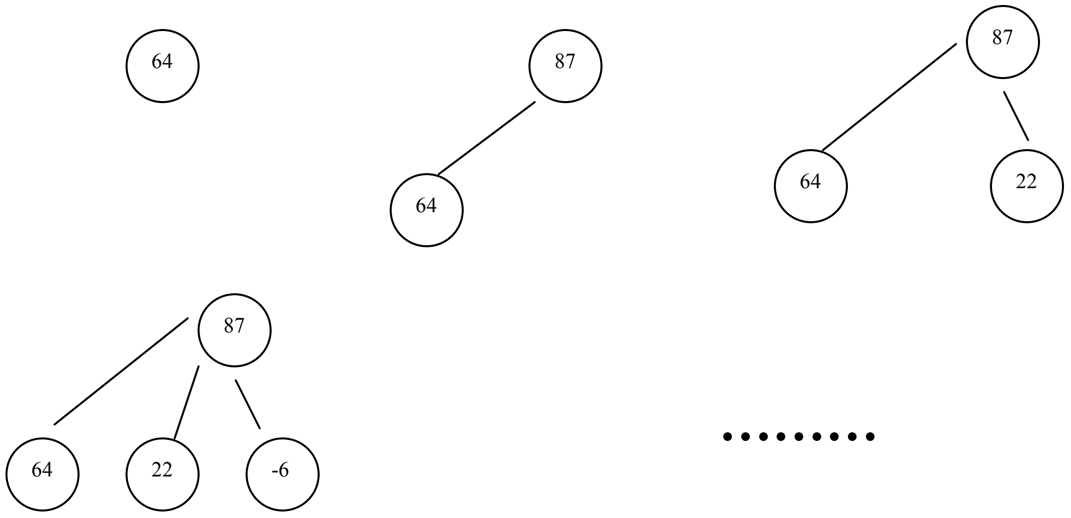
Solution (a)

A *d*-ary heap can be represented in a 1-dimensional array as follows. The root is kept in A[1], its *d* children are kept in order in A[2] through A[d + 1], their children are kept in order in A[d + 2] through A[$d^2 + d + 1$], and so on. The two procedures that map a node with index *i* to its parent and to its j^{th} child (for $1 \le j \le d$), respectively, are:

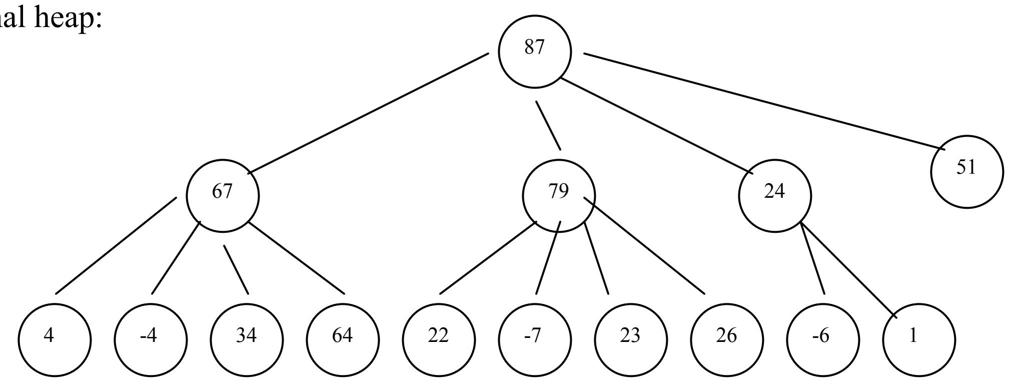
```
D-ARY-PARENT(i)
  return $\left\lfloor \frac{i-2}{d} + 1 \right\rfloor$

D-ARY-CHILD(i,j)

return $d(i-1) + j + 1$ where $1 \le j \le d$
```

To convince yourself that these procedures really work, verify that
D-ARY-PARENT(D-ARY-CHILD(*i*,*j*)) = *i*, for any $1 \le j \le d$.
Notice that the binary heap procedures are a special case of the above procedures when $d = 2$.

(b) Inserting 64, 87, 22, -6, 51, 4, -4, 34, 67, 79, -7, 23, 26, 1, 24



Final heap:


87
67
79
24
51
4
-4
34
64
22
-7
23
26
-6
1