COMP 122: Algorithms and Analysis

Practice Midterm Exam

October, 2005

Note: The following instructions are the same as those on the exam. This practice test has more questions than the exam (to give you more practice). The point values are intended to be representative of how much a comparable problem might be worth on the exam.

1. This exam contains 5 problems. You have 75 minutes to earn 100 points.
2. This exam is closed book. You may use one 8.5" x 11" crib sheet, with text on both sides if desired. The sheet must be a single sheet of paper, with nothing taped, stapled, or glued on it. When the exam is over, staple your crib sheet to the back of the exam.
3. You may use a calculator, but it shouldn't be necessary.
4. Do not spend too much time on any one problem. Read them all through first and attack them in the order that allows you to make the most progress.
5. Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Be neat.

Problem 1. (20pt): Solve the following recurrences. Do not worry about floors or ceilings. Justify your answers by naming the particular case of the master theorem, by iterating the recurrence, or by using the substitution method. If you use the substitution method, you only need to show an upper or lower bound, but you must be clear about which bound you are showing.
(a) $T(n) = T(n-2) + 1$

(b) $T(n) = 2T(n/2) + n \lg^2 n$

(c) $T(n) = 9T(n/4) + n^2$

(d) $T(n) = 3T(n/2) + n$

(e) $T(n) = T(n/2 + \sqrt{n}) + n$

Problem 2. (20pt): List the following functions in increasing asymptotic order. If two functions have equal asymptotic growth rate, then indicate this. You can use the notation $f(n) \prec g(n)$ to mean that $f(n) \in o(g(n))$, and $f(n) \sim g(n)$ to mean that $f(n) \in \Theta(g(n))$. Justify your answers, showing any algebraic simplifications you perform to make your comparisons.

$$n^2(\lg\lg n)^2 \qquad \sqrt{n} \qquad (\log_4 n)^3 \qquad n^2 \lg n \qquad 4^{\lg n} \qquad \lg(n^5)$$

Problem 3. (20pt): Answer the following questions:
(I) The kth *quantiles* of a set are the $k-1$ order statistics that divide the sorted set into k equal-sized sets (to within 1). Give an $O(n \lg k)$-time algorithm to list the kth quantiles of a set.
(II) Describe an algorithm that, given n integers in the range 1 to k, preprocesses its input and then answers any query about how many of the n integers fall into a range $[a..b]$ in $O(1)$ time. Your algorithm should use $O(n+k)$ preprocessing time.

Problem 4. (20pt):
A hash table of size m is used to store n items, with $n \leq m/2$. Open addressing is used for collision resolution.

(a) Assuming uniform hashing, show that for $i = 1, 2, \ldots, n$, the probability that the ith insertion requires srictly more than i probes is at most 2^{-i}.

(b) Show that for $i = 1, 2, \ldots, n$, the probability that the ith insertion requires more than $2 \lg n$ probes is at most $1/n^2$.

Let the random variable X_i denote the number of probes required by the ith insertion. You have shown in part (b) that $\Pr\{X_i > 2 \lg n\} \leq 1/n^2$. Let the random variable $X = \max_{1 \leq i \leq n} X_i$ denote the maximum number of probes required by any of the n insertions.

(c) Show that $\Pr\{X > 2 \lg n\} \leq 1/n$.

(d) Show that the expected length $E[X]$ of the longest probe sequence is $O(\lg n)$.

Problem 5. (20pt): Consider the following sorting algorithm called SILLYSORT.

At the beginning, array A has the input. SILLYSORT(A, i, j) sorts the elements in A from index i through index j.

Algorithm SILLYSORT(A, i, j):
Step 1.
Check if $A[i..j]$ is already sorted. If so, output and quit.
Step 2.
If $A[i] < A[i+1]$ then
SILLYSORT$(A, i+1, j)$
SILLYSORT(A, i, j)
Else
SILLYSORT$(A, i+1, j)$
Move $A[i]$ to the tail of list A, shifting all other elements one space to the left;
SILLYSORT(A, i, j)

Prove that the above algorithm correctly sorts the given n keys.