UNIVERSITY OF WATERLOO
School of Computer Science

CS860 **Adaptive Analysis** **Spring 2007**

ASSIGNMENT 1

(Due: Tuesday June 5, 2007, 2:30 PM)

Problem 1 (5 marks)
Consider the school book (classic) method of multiplying two numbers. Propose a natural measure of difficulty and give an adaptive analysis of the time taken by the multiplication algorithm above in terms of your proposed measure of difficulty.
Hint: Perform the multiplication in binary.

Problem 2 (4+6=10 marks)
Consider an airline company that keeps a database of co-pilots ranked by flight hours. Most of the time one of two operations are performed either (1) the highest ranked co-pilot gets promoted to captain and is removed from the database or (2) a new co-pilot gets hired and inserted in the database with full credit for past experience in other jobs. Now, at the end of every quarter the company has a policy of giving the most senior co-pilots a bonus.

We know that the database was implemented as a heap using an array.

a. One way to determine which pilots will get a bonus is to choose a threshold value T, and give a bonus to the set S of pilots in the database with T or more flight-hours experience. Give an algorithm that outputs the set S of pilots in time $O(h)$ where $h = |S|$.

b. Another way to select the top pilots is to give a bonus to the top k pilots only. Clearly we can produce the top k elements in the seniority list in $O(k \lg n)$ time using standard priority queue operations (deleteMax, insert). Describe a method that can determine the top k elements in $O(2^k + k^2)$ time. What is the largest value of k for which this solution beats the $O(k \lg n)$ algorithm?

Problem 3 (2+3=5 marks)
Combine what you learned in class about $2D$ convex hull and adaptive sorting.

a. One can sort a set of values $S = \{x_1, \ldots, x_n\}$ using a convex hull algorithm for points in two dimensions. Using any convex function such as $f(x) = x^2$, the convex hull of the set of points $\{\forall x \in S, (x, f(x))\}$ indicates the order of the values of S.

   Using this reduction, is there a straightforward adaptive sorting algorithm from one of the adaptive algorithms to compute the convex hull of points in two dimensions seen in class?

b. One can compute the upper part of the convex hull of points in two dimensions using a combination of sorting and divide and conquer: sort the points by their x-coordinate, divide into left and right part, compute the convex hull of each half, and merge the two in $O(\lg n)$ time (note that this algorithm is not adaptive in itself).

Can you deduce an adaptive convex hull algorithm from one of the adaptive sorting algorithms seen in class?

Problem 4 (2+2+2+2+2=10 marks)
We have seen in class various analyses of the intersection problem:

- *Gap Encoding* ($\mathcal{G}$) the difficulty measure introduced by Demaine, López-Ortiz and Munro, for with the algorithm "Adaptive" is optimal;
- *Alternation* (δ), the difficulty measure introduced by Barbay and Kenyon, for which the algorithm "Sequential" is optimal;
- *Redundancy* (ρ), the difficulty measure introduced by Barbay later on, for which the algorithm "Randomized" is optimal.

a. Any adaptive algorithm asymptotically optimal for the redundancy analysis is also optimal for the alternation analysis.

b. Prove that the reverse is not true: an adaptive algorithm which is asymptotically optimal for the alternation analysis is not always optimal for the redundancy analysis.

c. Consider an arbitrary adaptive algorithm which is asymptotically optimal for the alternation analysis. Is it necessarily optimal for the gap encoding analysis?

d. Consider an arbitrary adaptive algorithm which is asymptotically optimal for the gap encoding analysis. Is it necessarily optimal for the alternation analysis?

e. Consider an arbitrary adaptive algorithm which is asymptotically optimal for the gap encoding analysis. Is it necessarily optimal for the redundancy analysis?

Hint: To prove that an algorithm optimal for one difficulty measure is not always optimal for another difficulty measure, show that one of the three intersection algorithm "Adaptive", "Sequential" or "Randomized" is a counterexample.

Miscellany: There are many difficulty measures for adaptive sorting, and their relationships has been similarly studied.