Honors Algorithms — G22.3520-010 Fall 2006 — Problem Set 2
Due: Oct. 11

1. Consider maintaining a collection of lists of items on which the following operations can be performed:
   (i) Given two lists \( L_1 \) and \( L_2 \), form their concatenation \( L \) (destroying \( L_1 \) and \( L_2 \) in the process).
   (ii) Given a list \( L \) and a positive integer \( k \), split \( L \) into two lists \( L_1 \) and \( L_2 \), where \( L_1 \) consists of the first \( k \) items of \( L \), and \( L_2 \) the rest (\( L \) is destroyed in the process).
   (iii) Report the first item in given list \( L \).
   (iv) Create a new list with one item.

   (a) Describe data structures and algorithms supporting these operations so that operations (i)–(iii) can be performed in time \( O(\log n) \) (where \( n \) is the length of \( L \)), and operation (iv) takes constant time.

   (b) Suppose we also want to an operation that reverses a given list \( L \). Show how this operation can be implemented in constant time, while the other operations can still be performed within the time bounds of part (a).

2. In our analysis of Quick Sort in class, we showed that if \( T_j \) is the sum of squares of the problem sizes at level \( j \) of the recursion tree, then \( E[T_j] \leq (2/3)^j n^2 \). Let \( H \) be the height of the recursion tree. Show that \( E[H] = O(\log n) \).

   Hint: use the formula \( E[H] = \sum_{j \geq 1} \Pr[H \geq j] \)

3. In our analysis of Quick Select in class, we showed that if \( N_j \) is the problem size at level \( j \), then \( E[N_j] \leq \alpha^j n \), where \( \alpha = \sqrt{2/3} \). Show that this bound in fact holds for \( \alpha = 3/4 \).

4. Consider the following recursive, probabilistic algorithm \( A \), which takes as input a finite set \( S \) of items.

   Algorithm \( A(S) \):
   
   if \( |S| \leq 1 \)
   return \((0, 0, 0)\)
   
   else
   let \( R \) be a randomly chosen subset of \( S \)
   \((v_1, v_2, v_3) \leftarrow A(R)\)
   \((w_1, w_2, w_3) \leftarrow A(S \setminus R)\)
   return \((\max\{v_1, w_1\} + 1, v_2 + w_2 + |S|, v_3 + w_3 + |S|^2)\)

   Let \((X_1, X_2, X_3)\) denote the output of \( A \) on input \( S \), and let \( n := |S| \). Show that \( E[X_1] = O(\log n) \), \( E[X_2] = O(n \log n) \), and \( E[X_3] = O(n^2) \).

5. You are given integers \( a_1, \ldots, a_n \), where each \( a_i \) is an \( \ell \)-bit integer. Using Karatsuba’s algorithm as a subroutine, show how to compute the product \( \prod_{i=1}^n a_i \) in time \( O((n\ell)^{\log_2 3}) \).

6. Suppose we have an abstract data type that represents sets of items. There are three operations: \texttt{init}(S), which initializes an empty set \( S \); \texttt{copy}(S, T), which creates a set \( S \) that is a copy of \( T \); and \texttt{insert}(S, a), which inserts the item \( a \) into the set \( S \).

   Design an algorithm for the following problem. The input is a list of \( n \) distinct items \( a_1, \ldots, a_n \). The output is a list of \( n \) sets \( S_1, \ldots, S_n \), where \( S_i = \{a_1, \ldots, a_n\} \setminus \{a_i\} \).

   Your algorithm is only allowed to use the operations \texttt{init}, \texttt{copy}, and \texttt{insert}, described above, and should use \( O(n \log n) \) such operations.
7. You are given $n$ items, and you want to determine if there are any duplicates among them.

(a) Show how to solve this problem in expected linear time (and linear space) using universal hashing (assume evaluating the hash function and comparing items take unit time, and make any other reasonable assumption about the hash functions, as convenient).

(b) Show how to solve this problem using only comparisons in time $O(n \log n)$ (assuming comparisons take unit time).

(c) Prove an $\Omega(n \log n)$ time lower bound for this problem in the comparison model.

You may wish to proceed as follows.

– For a permutation $\pi$ on $\{1, \ldots, n\}$, let $v(\pi)$ denote the leaf in the decision tree reached on any input $(a_1, \ldots, a_n)$ satisfying $a_{\pi(1)} < a_{\pi(2)} < \cdots < a_{\pi(n)}$.

– Show that $v(\pi) \neq v(\pi')$ if $\pi \neq \pi'$.

Hint: for any permutation $\pi$, the comparisons along the path from the root to $v(\pi)$ naturally define a partial ordering on $\{1, \ldots, n\}$, and any partial ordering can always be extended to (at least) one total ordering.