Data Structures and Algorithms: Jan—Apr 2015

Midterm Exam

Instructions. There are five questions. All problems have equal weightage. When algorithms are asked, give a brief, but complete, description (in pseudocode form or in the form of steps) and a brief (in one or two sentences) proof of correctness too.

1. **Solve any *one* of (a) or (b).**
   (a) Given a list of *distinct* elements in non-decreasing order, give an algorithm to find, if any, an index i such that $A[i] = i$ in $O(\log n)$ time. Prove an $\Omega(n)$ lower bound for the problem, if the distinctness assumption is removed.
   (b) The *Hadamard matrices* $H_0, H_1, H_2, \ldots$ are defined as follows:
   - H_0 is the 1×1 matrix $[1]$.
   - For $k > 0$, H_k is the $2^k \times 2^k$ matrix:
   $$H_k = \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix}$$
   Show that if v is a column vector of length $n = 2^k$, then the matrix-vector product $(H_k)v$ can be calculated using $O(n \log n)$ operations. Assume that all the numbers involved are small enough that basic arithmetic operations like addition and multiplication take unit time.

2. Let $M(m, n)$ be the optimal number of comparisons to merge two sorted lists, one of size m and one of size n. Assume all elements are distinct.
   Solve any *one* of (a) or (b).
   (a) Show that $m + n - 1 \geq M(m, n) \geq \lceil \log \binom{m+n}{n} \rceil$
   (b) Show that when $M(n, n) \geq 2n - 1$ (using an adversary argument). Does the lower bound hold even for the case when m is not equal to n?

3. Given a balanced binary search tree (BST), suppose we want to support the operations $find(k, S)$ which returns the k-th smallest element (for $1 \leq k \leq n$) in the set S of keys stored in the tree. How would you augment the balanced BST to support this operation in $O(\log n)$ time? Argue that this augmentation can be maintained in $O(\log n)$ time under insertions, deletions and search.

4. Recall that an instance of the SET COVER problem consists of a ground set U, and a family $\mathcal{F}$ of subsets of U. The problem asks for a subfamily of the smallest size whose union is the ground set. In other words, we would like to minimize $|\mathcal{H}|$ where $\mathcal{H} \subseteq \mathcal{F}$ is such that $\bigcup_{S \in \mathcal{H}} S = U$. Show that for any integer n that is a power of 2, there is an instance of SET COVER with the following properties:
   (a) There are n elements in the ground set.
   (b) The optimal cover uses just two sets.
   (c) The greedy algorithm picks at least $\log n$ sets.

5. **Solve any *one* of (a) or (b).**

(a) Given a string X of size n, a *subsequence* of X is any string that is of the form

$$X[i_1]X[i_2]\cdots X[i_k], i_j < i_{j+1} \text{ for } j = 1, \ldots, k;$$

that is, it is a sequence of characters that are not necessarily contiguous but are nevertheless taken in order from X. For example, the string AAAG is a subsequence of the string CGATAATTGAGA.

Given two character strings, X of size n and Y of size m, over some alphabet, describe an algorithm that finds a longest string S that is a subsequence of both X and Y.

(b) Devise an algorithm that takes a sequence $x[1 \ldots n]$ and returns the (length of the) longest palindromic subsequence – a sequence that reads the same both in the forward and backward direction. Its running time should be $O(n^2)$.