Due Date: April 6, 2007

Problem 1: Let \( L = \{L_1, \ldots, L_n\} \) be a set of \( n \) lists, each containing \( 2^p \) items and each sorted in increasing order. Assuming that given \( 1 \leq i \leq n \) and \( 1 \leq j \leq 2^p \), one can find the \( j \)th item of \( L_i \) in \( O(1) \) time, describe an \( O(np) \)-time algorithm to compute the median of \( \bigcup_i L_i \). (Hint: Split each list into two, prune some of the lists that do not contain the median, and recurse.)

Problem 2: An \( m \times n \) matrix \( A = (a_{i,j}) \), \( 1 \leq i \leq m \) and \( 1 \leq j \leq n \) is called totally monotone if for all \( 1 \leq i_1 < i_2 \leq m \) and \( 1 \leq j_1 < j_2 \leq n \),

\[
a_{i_1,j_1} < a_{i_1,j_2} \Rightarrow a_{i_2,j_1} < a_{i_2,j_2}.
\]

(i) Describe an \( O((m + n) \log(m + n)) \)-time algorithm to find the maximum element in \( A \).

(ii) Assuming \( n \geq m \), describe an \( O(n) \) time algorithm to discard \( n - m \) columns that do not contain maximum element. (Hint: Compare \( A[k,k] \) and \( A[k,k+1] \) to determine whether one of the two columns can be discarded.)

(iii) Describe an \( O(m + n) \) time algorithm to find the maximum element in \( A \). (Hint: If \( n > m \), apply the above step to reduce the number of columns. Solve the problem recursively for the even rows.)

Problem 3: Let \( T = (V, E) \) be a (weighted) tree with \( n \) vertices in which each edge has a positive weight. For two vertices \( u, v \in V \), let \( d(u,v) \) be the total weight of the edges on the path in \( T \) from \( u \) to \( v \). Given an integer \( k \geq 1 \), the \( k \)-center of \( T \) is a set \( A = \{v_1, \ldots, v_k\} \) of \( k \) points so that

\[
\max_{u \in V} \min_{v \in A} d(u,v)
\]

is minimized.

(i) Describe a linear algorithm to compute a 1-center of \( T \).

(ii) Describe an \( O(n^2 \log n) \) algorithm to compute a \( k \)-center of \( T \) for any given \( k \).

(iii) \((*)\) Describe an \( O(n \log^c n) \)-time algorithm to compute a \( k \)-center of \( T \) for any given \( k \).