CSE 3500 Algorithms and Complexity
Exam III; December 4, 2014

Note: You are supposed to give proofs to the time and processor bounds of your algorithms. Read the questions carefully before attempting to solve them.

1. (16 points) Input are two sets $A$ and $B$ (not necessarily in sorted order) with $n$ elements in each. The problem is to check if these sets have any common elements. Present an $O(1)$ time $n^2$ processor common-CRCW PRAM algorithm for this problem.
2. (17 points) Input is a sequence of $n$ keys $k_1, k_2, \ldots, k_n$. The problem is to find the right neighbor of each key in sorted order. For instance if the input is 5, 2, 7, 2, 11, 15, 13, then the output will be 7, 11, 5, 2, 13, $\infty$, 15. How will you solve this problem in $O(1)$ time using $n^3$ common-CRCW PRAM processors?
3. (17 points) Input is a sequence $X = b_1, b_2, \ldots, b_n$ of bits. The problem is to find the position of the leftmost 1. For example, if $X = 0, 0, 0, 1, 0, 0, 1, 1$, the answer is 4. Present an $O(\log n)$ time algorithm to solve this problem. You can use up to $\frac{n}{\log n}$ CREW PRAM processors.
4. (16 points) Input are two strings $T = t_1 t_2 \cdots t_n$ (called the text) and $P = p_1 p_2 \cdots p_m$ (called the pattern) from some alphabet $\Sigma$. Assume that $m << n$. The problem is to output an array $A[1 : (n - m + 1)]$ such that $A[i] = d_i$ where $d_i$ is the Hamming distance between $t_i t_{i+1} \cdots t_{i+m-1}$ and $P$, for $1 \leq i \leq (n - m + 1)$. Hamming distance between two strings $A$ and $B$ of equal length is the number of positions in which they differ. For example, if $A = aababaabbab$ and $B = abbbabaabab$, then the Hamming distance between $A$ and $B$ is 6. Present an $O(\log m)$ time algorithm for this problem. You can use up to $\frac{mn}{\log m}$ CREW PRAM processors.
5. (17 points) \( \pi_1 \) and \( \pi_2 \) are two decision problems. It is known that \( \pi_1 \preceq \pi_2 \). Any instance of \( \pi_1 \) can be translated into an instance of \( \pi_2 \) in \( n^3 \) time such that the instance of \( \pi_1 \) has the answer yes iff the corresponding instance of \( \pi_2 \) has the answer yes. Here \( n \) is the size of the instance of \( \pi_1 \). If \( \pi_2 \) can be solved in time \( O(m^{10}) \) on any instance of size \( m \), how fast can \( \pi_1 \) be solved using the above reduction on an instance of size \( n \)?
6. (17 points) The traveling salesman problem (TSP) is defined as follows: Input is a directed weighted graph $G(V, E)$ and a real number $t$. The edge weights can be arbitrary non-negative real numbers. The problem is to decide if there exists a tour of total weight $\leq t$. A tour is nothing but a path that starts from a node, visits every node exactly once and comes back to the starting node. The weight of a tour is nothing but the sum of all the edge weights along this path.

The Hamiltonian cycle problem (HCP) is defined as follows: Input is a directed graph $G(V, E)$. The problem is to check if there is a directed cycle in the graph in which each node of $G$ occurs exactly once.

Show that HCP polynomially reduces to TSP.