CSE361 Computational Complexity of Algorithms. Fall 2005

Home work 4. Due on December 1 (2 PM).

1. Show how to multiply two $n \times n$ matrices using $\frac{n^3}{\log n}$ CREW PRAM processors and $O(\log n)$ time.

2. The string matching problem takes as input a text t and a pattern p, where t and p are strings from an alphabet Σ. The problem is to determine all the occurrences of p in t. Present an $O(1)$-time PRAM algorithm for string matching. Which PRAM are you using and what is the processor bound of your algorithm?

3. 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 $5.2, 7, 2, 11, 15, 13,$ the output is $7, 11, 5.2, 13, \infty, 15$. How will you solve this problem in $O(1)$ time using n^3 CRCW PRAM processors?

4. Show that the knapsack optimization problem reduces to the knapsack decision problem when all the p's, w's, and m are integers and the complexity is measured as a function of input length. (**Hint**: If the input length is q, then $\sum p_i \leq n2^q$, where n is the number of objects. Use a binary search to determine the optimal solution value.)

5. Let π_2 be a problem for which there exists a deterministic algorithm that runs in time $2^{\sqrt{n}}$ (where n is the input size).

   Prove or disprove: "If π_1 is another problem such that π_1 is polynomially reducible to π_2, then π_1 can be solved in deterministic $O(2^{\sqrt{n}})$ time on any input of size n."

6. Let $F = \{S_j\}$ be a finite family of sets. Let $T \subseteq F$ be a subset of F. T is a cover of F iff $\bigcup_{S \in T} S = \bigcup_{S \in F} S$. The set cover decision problem is to determine whether F has a cover T containing no more than k sets. Show that the node cover decision problem is polynomially reducible to this problem.