CSE 361 Complexity of Sequential and Parallel Algorithms

Spring 2008; Home work 3. Due on April 28 (3:30 PM).

1. Present an $O(n)$ time algorithm to compute the coefficients of the polynomial $(1+x)^n$. How much time is needed if you use the FFT algorithm to solve this problem?

2. An $n \times n$ *Toeplitz* matrix is a matrix A with the property that $A[i,j] = A[i-1, j-1],\ 2 \leq i, j \leq n$. Give an $O(n \log n)$ algorithm to multiply a Toeplitz matrix with an arbitrary $(n \times 1)$ column vector.

3. Let $f(x)$ be a polynomial of degree $n > 0$. This polynomial has n derivatives, each one obtained by taking the derivative of the previous one. Devise an algorithm that computes all the derivatives of $f(.)$ at a given point a. Your algorithm should run in time $O(n \log^2 n)$.

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.

7. Present a common CRCW PRAM algorithm that finds the maximum of n arbitrary elements in $O(1)$ time using $n^{1+\epsilon}$ processors for any fixed $\epsilon > 0$.

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

9. 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?