CS 1510 Midterm 2
Fall 2010

1. (20 points) Consider the following two problems:

   TRAVELING SALESMAN

   INPUT: points $(x_1, y_1), \ldots (x_n, y_n)$ in the Euclidean plane

   OUTPUT: The shortest route, starting from the origin, that visits all the points. You can assume that the route is specified by the order that the points are visited.

   SORTING

   INPUT: Numbers $z_1, \ldots, z_k$

   OUTPUT: These numbers in increasing order

   Assume that SORTING is a well known problem for which no linear time algorithm is known. Explain/Prove, using the concept of a reduction, that finding a linear time algorithm for TRAVELING SALESMAN is just as hard as finding a linear time algorithm for SORTING.

   The purpose of this problem is mostly to test whether you understand what a reduction is, and how one can use a reduction to relate the relative hardness of problems. To receive full credit, your explanation must be reasonably understandable by one of your fellow computer science majors who has not taken this class.

2. (a) (5 points) Explain how to solve the Parallel Prefix Problem on an EREW PRAM in time $O(\log n)$ on input of size n with n processors. Start with defining the Parallel Prefix Problem.

   (b) (5 points) What is the efficiency of the algorithm in the previous subproblem? Start with a definition of efficiency. You need not have solved the previous subproblem to answer this question.

   (c) (5 points) What would the Folding Principle say about the time for the algorithm in the first subproblem if there were only $n^{1/3}$ processors? Start with a definition of the Folding Principle in terms of the time taken by a parallel algorithm. You need not have solved the first subproblem to answer this question.

   (d) (5 points) State the folding principle in terms of the efficiency of a parallel algorithm. Show that this definition of the folding principle in terms of efficiency is equivalent to the definition of the folding principle in terms of time that you gave in the previous subproblem.

3. You may answer at most one of the three subproblems. The subproblems are worth different amounts of points, and are of different difficulties. You must clearly identify which subproblem you are answering.

   (a) (5 points) Show by reduction that if the problem of determining whether a graph G has an independent set of a specified size k has a polynomial time algorithm then the problem of determining whether a graph H has an clique of a specified size ℓ has a polynomial time algorithm. Recall that a clique is a collection of mutually adjacent vertices, and an independent set is a collection of mutually nonadjacent vertices.

   (b) (10 points) Consider the following problem. The input is an undirected graph G and an integer k. The problem is to determine if G contains a clique of size k AND an independent set of size k. Show by reduction that if this problem has a polynomial time algorithm then the clique problem has a polynomial time algorithm (recall that the Clique problem is to determine whether a graph has a clique of a specified size).

   (c) (20 points) Consider the following problem. The input is an undirected graph G and an integer k. The problem is to determine if G contains a clique of size k OR an independent set of size k. Show by reduction that if this problem has a polynomial time algorithm then the clique problem has a polynomial time algorithm.

4. You may answer at most one of the three subproblems. The subproblems are worth different amounts of points, and are of different difficulties. You must clearly identify which subproblem you are answering.

   (a) (5 points) Consider the following problem. The input is n numbers stored in a doubled linked list. Assume that each of n processors has a pointer to a unique arbitrary item in the linked list. The output is an array with these n numbers stored in the same order as they were in the linked list. Give a parallel algorithm for this problem that runs in time $O(\log n)$ on an EREW PRAM with n processors.

   (b) (10 points) Consider the following problem. The input is a binary tree with n nodes. Assume that each of n processors has a pointer to a unique arbitrary node in the tree. The output should be an array storing the leaves of the tree, that are right children of their parents, in the order that they would be visited by an in-order traversal of the tree (intuitively in left to right order if you drew the tree). Give a parallel algorithm for this problem that runs in time $O(\log n)$ on an EREW PRAM with n processors.

   (c) (20 points) Design a parallel algorithm that takes as input a binary expression tree, where the leaves are Boolean values 0 or 1, and each internal node is one of the three standard logical operations: NOT, OR, and AND. The output should be the value of the expression represented by the tree. Your algorithm should run in $O(\log^2 n)$ time on a CREW PRAM with n processors, where n is the number of nodes in the tree. You may assume that each processor initially has a pointer to a unique arbitrary node in the tree.

5. Consider the following problem:

   MERGING

   INPUT: Two arrays A and B of numbers, each of size n, and each sorted in increasing order

   OUTPUT: An array C of size $2n$ that contains exactly the same numbers as A and B, sorted in increasing order.

   You may answer at most one of the three subproblems. The subproblems are worth different amounts of points, and are of different difficulties. You must clearly identify which subproblem you are answering.

   (a) (5 points) Give a parallel algorithm that runs in time $O(\log n)$ on a EREW PRAM with n processors.

   (b) (10 points) Give a parallel algorithm that runs in time $O(1)$ on a CRCW Common PRAM with n^2 processors.

   (c) (20 points) Give a parallel algorithm that runs in time $O(\log \log n)$ on a CRCW Common PRAM with n processors.