CS 1510 Midterm 2
Fall 2010

The test is a bit long. I suggest keeping your answers short and to the point.

1. (a) (5 points) Consider the standard EREW algorithm for adding n integers that runs in time $T(n,p) = n/p + \log p$ with p processors. Define the "efficiency" of a parallel algorithm. Draw a rough curve showing the efficiency of this algorithm as a function of the number of processors p. Put p on the horizontal axis, and the efficiency of this algorithm on the vertical axis. Identify for which p the efficiency of the algorithm will be at least a constant (technically $\Omega(1)$).

   (b) You have an EREW algorithm A that runs in time $n^{1/2}$ with $p = n^{2/3}$ processors. Give upper and lower bounds on the running time for A with $n^{1/4}$ processors. Justify your answers.

   (c) Recall that the input to the subset sum problem was positive integers $x_1, \ldots, x_n, L$. We know that there is a dynamic programming algorithm for this problem that runs in time $O(nL)$. Is this a polynomial time algorithm? Justify your answer, starting with a careful definition of "polynomial time algorithm".

   (d) (5 points) An excited CS 1 student claims that they have a program to solve the Halting Problem. They give you a program $IAMFAMOUS(P, I)$ that takes as input a string P representing a program, and another string I. The claimed output of $IAMFAMOUS$ is 1 if the program P halts on the input I, and 0 otherwise. Give an example of a program P and a string I on which the program $IAMFAMOUS$ will output the incorrect answer. Hint: Your answer may incorporate the code for $IAMFAMOUS$ that is provided to you by the enthusiastic CS 1 student.

2. (20 points) Consider the following two problems:

   VERTEX COVER OPTIMIZATION

   INPUT: A graph G

   OUTPUT: A vertex cover of G that contains the least number of vertices.

   VERTEX COVER DECISION

   INPUT: A graph H and an integer ℓ

   OUTPUT: 1 if H contains a vertex cover of size ℓ and 0 otherwise.

   Recall that a vertex cover is a collection S of vertices such that every edge is incident on at least one vertex in S. Show that the vertex cover problem is self-reducible. Start with a definition of "self-reducible".

3. (20 points) Consider the following problem. The input is a graph $G = (V, E)$, a subset R of vertices of G, and a positive integer k. The problem is to determine if there is a subset U of V such that

   - All the vertices in R are contained in U, and
   - the number of vertices in U is at most k, and
   - for every pair of vertices x and y in R, one can walk from x to y in G only traversing vertices that are in U.

   Show that this problem is NP-hard using a reduction from Vertex Cover. Recall that the input for the vertex cover problem is a graph H and an integer ℓ, and the problem is to determine whether H has a vertex cover of size ℓ or not. A vertex cover S is a collection of vertices with the property that every edge is incident on at least one vertex in S.

   Make sure to explain the general set up of how one reduces one problem to another problem in addition to the details specific to this reduction, and make sure not to mess up the direction. You can earn partial credit for this, even if you don't specifically how to do the reduction.

4. (20 points) Give an algorithm to sort n numbers in time $O(\log^2 n)$ with n processors on an EREW machines. Give a couple of sentences of intuition why the algorithm is correct, and why the running time is $O(\log^2 n)$; You don't need a fully formal proof.

5. (20 points) Explain how to simulate a parallel algorithm A that runs in time $O(T(n, p))$ on an CREW machine with p processors by a parallel algorithm B that runs in time $O(T(n, p) * \log^2 p)$ on an EREW machine with p processors.