CS 1510 Midterm 2
Fall 2005

1. (a) Define EREW. That is, what properties must a PRAM program have to be EREW.
   (b) Explain how to merge two sorted lists $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ of integers in time $O(\log n)$ on an EREW PRAM using $p = n$ processors.
   (c) State the efficiency of the algorithm in part b. Start with a definition of efficiency. You can answer this question even if you did not answer part b.
   (d) Explain how to sort a sorted list $x_1, \ldots, x_n$ of integers in time $O(\log^2 n)$ on an EREW PRAM using $p = n$ processors.

2. Assume that we know that the problem, which we call the X problem, of determining whether a graph G has a subgraph H, where H has at least some integer k number of vertices and H has property X, is NP-complete. Now consider the Y-problem of determining whether a graph G has a subgraph H, where H has at least some integer k number of vertices and H has property Y. Note that we do not, for now, specify the properties X and Y.

   (a) (10 points) Explain how you would prove that the Y problem is NP-hard, using the fact that the X poblem is NP-complete. Be as precise and complete as possible.
   (b) (10 points) Now assume that the property X is that the subgraph H is a clique, that is, all the vertices in H are adjacent. Further assume that the property Y is that the subgraph H is an independent set, that is, none of the vertices in H are adjacent. Explain how to show that the Independent Set problem is NP-hard using the fact that the Clique problem is NP-complete. You need not repeat your answer from part a. You need only specifying the part of the answer that depends on the exact natures of the properties X and Y.

3. The input to this problem is a character string C of n letters. The problem is to find the largest $k < n/2$ such that

$$C[k]C[k-2]\ldots C[1] = C[n-k+1]\ldots C[n-1]C[n]$$

   That is, k is the length of the longest prefix that is also a suffix reversed. So for example, if the input was $C = cabbaxyzabbac$, then the answer would be $k = 5$ since the prefix *cabba* of length 5 is the reverse of the prefix *abbac*. Note that that this problem is similar, but not identical, to one of the homework problems. Give a EREW parallel algorithm that runs in $O(\log n)$ with $p = n$ processors.

4. Show that the knapsack problem is self-reducible. The input to the decision problem is a collection of objects with positive integer values $v_1, \ldots, v_n$, weights $w_1, \ldots, w_n$, weight limit W, and value goal V. The decision problem is to determine if there is a subset of the objects with aggregate weight no more than W, and aggregate value at least V. The input to the optimization problem is a collection of objects with positive integer values $v_1, \ldots, v_n$, weights $w_1, \ldots, w_n$, and weight limit W. The optimization problem asks you to return the collection of objects, with aggregate weight at most W, and with maximum aggregate value.

5. The input to this problem is n points $x_1, \ldots, x_n$ on a line. A good path P has the property that one endpoint of P is the origin and every x_i is covered by P. Note that P need not be simple, that is, it can backtrack over territory that it has already covered. Assume a vehicle moves along this path from the origin at unit speed. The response time r_i for each x_i is the time until the vehicle first reaches x_i. The problem is to find the good path that minimizes $\sum_{i=1}^{n} r_i/n$, the average response time. For example, if the points are $x_1 = 1$ $x_2 = 8$ and $x_3 = -2$ and the path visited the points in the order x_1, x_3, x_2, the average response time for this path would be $1/3 + (1+3)/3 + (1+3+10)/3$. Give a polynomial time algorithm for this problem.