CS 473U: Undergraduate Algorithms, Fall 2006
Homework 8

Due Wednesday, December 6, 2006 in 3229 Siebel Center

Remember to submit **separate, individually stapled** solutions to each of the problems.

---

1. Given an array $A[1..n]$ of $n \geq 2$ distinct integers, we wish to find the second largest element using as few comparisons as possible.

   (a) Give an algorithm which finds the second largest element and uses at most $n + \lceil \lg n \rceil - 2$ comparisons in the worst case.

   ⋆(b) Prove that every algorithm which finds the second largest element uses at least $n + \lceil \lg n \rceil - 2$ comparisons in the worst case.

2. Let R be a set of rectangles in the plane. For each point p in the plane, we say that the *rectangle depth* of p is the number of rectangles in R that contain p.

   (a) (Step 1: Algorithm Design) Design and analyze a polynomial-time algorithm which, given R, computes the maximum rectangle depth.

   (b) (Step 2: ???) Describe and analyze a polynomial-time reduction from the maximum rectangle depth problem to the maximum clique problem.

   (c) (Step 3: Profit!) In 2000, the Clay Mathematics Institute described the Millennium Problems: seven challenging open problems which are central to ongoing mathematical research. The Clay Institute established seven prizes, each worth one million dollars, to be awarded to anyone who solves a Millennium problem. One of these problems is the P = NP question. In (a), we developed a polynomial-time algorithm for the maximum rectangle depth problem. In (b), we found a reduction from this problem to an NP-complete problem. We know from class that if we find a polynomial-time algorithm for any NP-complete problem, then we have shown P = NP. Why hasn't Jeff used (a) and (b) to show P = NP and become a millionaire?

3. Let G be a complete graph with integer edge weights. If C is a cycle in G, we say that the *cost* of C is the sum of the weights of edges in C. Given G, the traveling salesman problem (TSP) asks us to compute a Hamiltonian cycle of minimum cost. Given G, the traveling salesman cost problem (TSCP) asks us to compute the cost of a minimum cost Hamiltonian cycle. Given G and an integer k, the traveling salesman decision problem (TSDP) asks us to decide if there is a Hamiltonian cycle in G of cost at most k.

   (a) Describe and analyze a polynomial-time reduction from TSP to TSCP.

   (b) Describe and analyze a polynomial-time reduction from TSCP to TSDP.

   (c) Describe and analyze a polynomial-time reduction from TSDP to TSP.

(d) What can you conclude about the relative computational difficulty of TSP, TSCP, and TSDP?

4. Let G be a graph. A set S of vertices of G is a *dominating set* if every vertex in G is either in S or adjacent to a vertex in S. Show that, given G and an integer k, deciding if G contains a dominating set of size at most k is NP-complete.