CS 577: Introduction to Algorithms 12/10/2015

Practice Problems for the Final Exam

Instructor: Dieter van Melkebeek TAs: Kevin Kowalski, Andrew Morgan, and Bryce Sandlund

1. In the two-player game "Two Ends", n cards are laid out in a row. On each card, face up, is written a positive integer. Players take turns removing a card from either end of the row and placing the card in their pile, until all cards are removed. The score of a player is the sum of the integers of the cards in his/her pile.

   Give an efficient algorithm that takes the sequence of n positive integers, and determines the score of each player when both play optimally.

2. You are given an arithmetic expression containing n integers and $n-1$ operators, each either $+$, $-$, or $\times$. You are also given a positive integer m. Your goal is to find out whether there is an order to perform the operations such that the result is a multiple of m.

   For example, for the expression $6\times3+2\times5$, this is possible for $m=7$, namely as follows: $(6\times3)+(2\times5) = 28$. For the same expression this is not possible for $m=8$ as none of the five possible orderings yield a multiple of 8: $(6\times3)+(2\times5) = 28$, $((6\times(3+2))\times5) = 150$, $((6\times3)+2)\times5 = 100$, $6\times(3+(2\times5)) = 78$, $6\times((3+2)\times5) = 150$.

   Design an algorithm that runs in time polynomial in n and m.

3. You are given an $n\times n$ table with entries in $\{0,1\}$. Determine the largest size of a contiguous square in the table that consists solely of 1s. Your algorithm should run in time $O(n^2)$.

4. You collect coupons and aim to have a collection containing each of the n types of coupons that exist. Your current collection consists of exactly n coupons but contains duplicates. You are hoping to achieve your goal by a sequence of exchanges of a coupon of one type for a coupon of another type, but only certain types of exchanges are possible.

   For example, suppose that your current collection consists of one coupon of type 1, and 3 coupons of type 2, and that it is possible to exchange a coupon of type 2 for a coupon of type 3, and a coupon of type 3 for a coupon of type 4. In this case, it is possible for you to achieve your goal, namely by exchanging two of your coupons of type 2 for coupons of type 3, and one of the obtained coupons of type 3 for a coupon of type 4.

   Develop an efficient algorithm that, given your current collection and the possible exchanges, determines whether your goal is achievable or not.

5. You would like to make a network of wireless sensors more reliable by selecting some number of "backup" sensors for each sensor in the network.

   More specifically, you are given the coordinates (x_i, y_i) of sensor s_i for $i = 1, \ldots, n$, and positive integers d, r, and $b \geq r$. Your goal is to find backup sets B_j for $j = 1, \ldots, n$ such that $B_j \subseteq \{s_1, \ldots, s_n\} - \{s_j\}$, every sensor in B_j is within distance d of s_j, every B_j contains exactly r sensors, and every sensor s_i for $i = 1, \ldots, n$ belongs to at most b sets B_j.

   Develop an efficient algorithm that realizes the goal or reports that it is impossible.

6. Give brief arguments showing that the following decision problems are NP-hard.

   (a) *Longest Path*: Given an unweighted directed graph G and a number k, does there exist a simple path in G (i.e., a path in which no vertex is repeated) with at least k edges.

   (b) *Max-SAT*: Given a CNF formula and a number k, is there an assignment to the variables that satisfies at least k clauses.

(c) *Dense-subgraph*: Given a graph and two integers a and b, is there a subset of vertices of size at most a that contain at least b edges between them.

7. Consider the following problem: Given an undirected graph $G = (V, E)$, find a subset $S \subseteq V$ such that $|\Gamma(S) \setminus S|$ is maximized. Recall that $\Gamma(v)$ for a vertex v denotes all of the vertices u such that $(u, v) \in E$, and that $\Gamma(S) = \cup_{v \in S} \Gamma(v)$.

   Formulate a decision version corresponding to this problem such that the decision version is equivalent to the given problem under polynomial-time reductions, and is NP-complete. Prove both properties.

8. You are given a graph G with non-negative integer edge weights, vertices s and t, and an integer ℓ, and would like to find a (not necessarily simple) path from s to t of length exactly ℓ, or report that none exists.

   (a) Give a pseudo-polynomial-time algorithm for this problem.

   (b) Show that the problem is NP-hard.