CSE 521

Design & Analysis of Algorithms I — Winter 2010

Take-Home Midterm — Instructor: Paul Beame

Due **Feb 23 at the beginning of class**

Instructions: This is a take-home exam with the following rules and instructions:

- The work you turn in should be entirely your own. You are not allowed to collaborate or discuss the problems, solutions, or any aspect relating to this exam with your classmates, or anyone else, except the course staff. You are also not allowed to use any written or electronic materials except for the Kleinberg-Tardos text and our course materials on the web.
- If you are having any difficulty understanding any of the questions or think that something is ambiguous, come talk to, make an appointment with, or send an email to the course staff.
- Each problem is worth the same amount.
- Devote sufficient time for writing your solutions in a clear manner. We are looking for correctness, precision, and clarity of exposition.
- Handwritten solutions are just fine but make sure that your solutions are very clearly organized and legible.
- You do not need to prove anything that is proven in the book or was proven in class. When you refer to something in the book, please refer to a page number.
- If you are not going to be able to make it to class on Feb 23, please arrange to turn in the exam ahead of time.

1. (Designer sequence cost) You are the operations manager for a designer sequence company. Your expertise is in building custom sequences using the letters $\{A, C, G, T\}$. You can do this by splicing together sequences from your storehouse of pre-built sequences $s_1, \ldots, s_m$ with lengths $\ell_1, \ldots, \ell_m$, respectively. You have an unlimited supply of each of these sequences but each time a sequence s_i is spliced in, it incurs an additional cost c_i. Your storehouse of sequences includes all the length 1 sequences so that you can construct any sequence your clients desire.

   (a) Design a polynomial-time algorithm that takes as input a sequence x as well as the full information about your storehouse of pre-built sequences and determines the least total cost of building x as a concatenation of pre-built sequences and argue that it is correct.

   (b) Describe how to find an optimal way to build x from these sequences.

   (c) Analyze the running time and space usage of your algorithm(s) as a function of m and n, the length of x.

2. (Best single investment) Consider a simplified stock market in which there is a single stock whose price changes only once per day. If one knew the daily sequence of prices $p_1, \ldots, p_n$ of a stock in advance then an optimal strategy would be to buy on every day when the price is at a local minimum and sell on every day when it is a local maximum. However this may require many separate investments.

   Suppose instead that though you know the price sequence you are restricted to making only one market investment during this time period. Derive a linear time algorithm to find the single pair (i, j) with $i \leq j$ such that buying on day i and selling on day j will result in the maximum profit. (If there is no profit to be made then choosing $j = i$ for a profit of 0 is an allowable single market investment that achieves the maximum.) Justify the correctness and running time of your algorithm.

3. (Fully-capacitated network flow.) In the basic network flow problem the input is a directed graph $G = (V, E)$ with two distinguished vertices s and t as well as a non-negative capacity $c(e) = c(u, v)$ for each directed edge $e = (u, v)$ in G. There are capacity constraints on the flow along each edge as well as conservation of flow constraints at each vertex v other than s and t. Sometimes, however, the vertices in a network represent some form of switch that itself has a limited capacity. In the *fully-capacitated network flow* problem, in addition to the edge capacities and conservation of flow we also have a non-negative capacity c_v for each vertex v other than s or t and, in addition to all the requirements of regular network flow, require that the total flow that passes through vertex v (which is $f^{in}(v) = \sum_{(u,v)} f(u, v) = f^{out}(v) = \sum_{(v,w)} f(v, w)$) must be at most c_v.
   Show how to extend the algorithms that you know to yield a polynomial-time algorithm for the fully-capacitated network flow problem, give its running time, and justify its correctness.

4. (01-Polynomial Equations) An integer polynomial in variables $x_1, \ldots, x_n$ is a finite sum of terms of the form $c_{i_1 \ldots i_n} \cdot x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ where each $i_1, \ldots, i_n$ [was i_m] is an integer ≥ 0 and each $c_{i_1 \ldots i_n}$ is an integer. The 01-Polynomial-Equations problem (01-PE) is the following: Given a sequence of integer polynomials $P_1(x_1, \ldots, x_n), \ldots, P_m(x_1, \ldots, x_n)$ does the system of equations

$$\begin{aligned} P_1(x_1, \ldots, x_n) &= 0 \\ P_2(x_1, \ldots, x_n) &= 0 \\ \ldots &= \ldots \\ P_m(x_1, \ldots, x_n) &= 0 \end{aligned}$$

   have a solution in which each x_i has a value in $\{0, 1\}$? Prove that 01-PE is NP-complete.

   Structure your proof so that the parts are clear.