CSC364 Summer 2004 — Homework 2

The following questions are assigned each week. You need only hand in those questions marked with a star * as part of your assignment, and only those questions will be marked. Make sure you include a completed and signed cover page for each problem.

Week 5

1. We tend to consider problems solved by polynomial time algorithms as tractable. What happens if we use such algorithms as procedures in loops, recursive calls, or different phases of our programs? Will such a program be tractable, or will the running time be worse than polynomial? The fundamental question is whether our idea of tractable is closed under such operations.

   Prove that the class of polynomials over real numbers, $\mathbb{R}[x]$, is closed under

   (a) addition,

   (b) multiplication, and

   (c) composition.

   That is, for polynomials $p(x)$ and $q(x) \in \mathbb{R}[x]$, prove that $p(x) + q(x)$, $p(x) \cdot q(x)$, and $p(q(x)) \in \mathbb{R}[x]$.

*2. (hand in) Let $S_1, S_2, \ldots, S_n$ be a collection of n MP3 songs. Song S_i requires m_i kilobytes of storage. We have a personal MP3 player with D kilobytes of memory where $D < \sum_{i=1}^{n} m_i$. We want to load the maximum number of different songs onto our MP3 player.

   (a) Give a greedy algorithm that outputs the set of songs to load such that we load the maximum number of songs possible.

   (b) Prove that your algorithm correctly finds the maximum number of songs we can store on the player.

   (c) State and prove the running time of your algorithm.

3. Consider the same MP3 player as in question 2. We want to use as much of the player memory as possible. Prove or disprove: we can use a greedy algorithm to select songs leaving the minimum amount of unused storage space on the player.

Week 6

4. Consider the problem of making change for n cents using as few coins as possible.

   (a) Suppose the available denominations of coins are powers of $d > 1$: there are $d^0, d^1, d^2, \ldots, d^k$ cent coins for some $k \geq 1$. Show that there is a greedy algorithm that always yields an optimal solution.

   (b) Give a set of coin denominations for which the greedy algorithm does not always yield an optimal solution. You should include the unit coin (penny) so that there is a solution for every value of n.

*5. (hand in) Consider a manufacturing process which must cut a steel beam into n pieces where $l_1, l_2, \ldots, l_n$ is the set of lengths for the pieces. That is, piece i should have length l_i. Assume that $\sum_{i=1}^{n} l_i$ is equal to the original length of the beam.

   Due to the specifications of the machine, the cost of cutting a beam segment into two pieces is $k + cl$ where l is the length of the beam segment and k, c are positive constants.

   Design an algorithm to determine an optimal order of cuts which minimizes the total cost. Prove your algorithm correct and justify its running time. *Note:* Higher marks to more efficient algorithms.

   Hint: It may be easier to first consider the reverse process of sticking each piece back together where the cost of each attachment is $k + cm$ where m is the sum of the lengths of the two segments being joined. Then reverse the order.

6. Suppose you have l sorted sequences $X_1, X_2, \ldots, X_l$ such that $\sum_{i=1}^{l} |X_i| = n$. Show how to combine the l sequences into a single sorted sequence in time $\mathrm{O}(n \log l)$.

Week 7

*7. (hand in) You are given the task of modeling acceleration in vehicles. A vehicle begins its acceleration in the first gear, and as it accelerates, it may shift to higher gears in order to increase the vehicle's speed. We will simplify the problem by discretizing it to 1 second time intervals. This means that you only need to consider $S(t)$ where S is the state of the vehicle at time t and t is an integer.

   For gear i, the function $f_i : \mathbb{R} \rightarrow \mathbb{R}$ models the acceleration of the vehicle in that gear. In particular, if the vehicle is in gear i with current speed s, then in 1 second, the vehicle will have speed $f_i(s)$.

   It takes 1 second to change to the next higher gear, and the vehicle will lose some speed during the gear change. The deceleration function $g : \mathbb{R} \rightarrow \mathbb{R}$ determines how much speed is lost for a gear change. If the vehicle is at speed s, then after the gear change, the vehicle will be at speed $g(s)$. We never change to a lower gear.

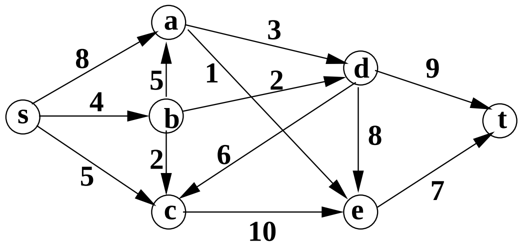
   The input to your algorithm is k, where k is the number of gears, k acceleration functions $f_1, \ldots, f_k$, a deceleration function g, and a time T. You are to efficiently determine the maximum speed the vehicle can achieve in T seconds if it begins from a complete stop. Prove the correctness of your algorithm and its running time.

8. Consider again the problem of making change for n cents using as few coins as possible. Design an efficient algorithm that always finds an optimal solution for any set of denominations. Prove you algorithm correct and justify its running time.

9. Suppose we are given an arbitrary arithmetic formula as a series of alternating operands and operators. For example, our sequence f might be $5+7\times6-2\div12$. We know that the value of f will depend on how we parenthesize the operations. Design an efficient algorithm which determines the necessary order of operations that *maximizes* the value of f. Give the running time of your algorithm and prove your algorithm correct.

Week 9

*10. (hand in) Consider the following network with source s, target t, and the given capacities for each arc.



Demonstrate Ford-Fulkerson's algorithm on this network. Show the augmenting path found at each step of the algorithm. Prove that the flow found by the algorithm is maximum in this network by demonstrating a cut of the same size.

11. (Problem 26-1.a of CLRS) Consider a flow in which the vertices, as well as edges, have capacities. That is, the total positive flow entering a given vertex is subject to a capacity constraint. Show that determining the maximum flow in a network with edge and vertex capacities can be reduced to an ordinary maximum-flow problem on a flow network of comparable size.

12. Bell Canada has a network of phone lines and switching stations. Each device (telephone, fax, computer, etc.) connected to the network is identified by a unique phone number. The capacity of a line or a switch is the number of calls that line or switch can handle simultaneously. Assume the capacity of the phone line from a device to the nearest switch is 1. Assume the capacity of switch i is s_i, and assume the capacity of line ij which connects switch i to switch j is l_{ij}.

    Suppose there are m phone calls that are to occur simultaneously between m unique initialing devices $(p_1, \ldots, p_m)$ to m receiving devices $(q_1, \ldots, q_m)$. Give an efficient algorithm which either finds a route through the phone network so that all m calls can occur simultaneously or reports that this is impossible. Prove the running time and justify the correctness of your algorithm.