Assignment 4

Algorithms: Correctness and Complexity

Due Nov 29, 2012

The work that you turn in for this assignment must represent your individual effort. You are welcome to help your fellow students understand the material of the course and the meaning of the assignment questions, however, the answer that you submit must be created by you alone.

Q0[30]. Dr Brown is driving his DeLorean from Sidney, N.S. to Vancouver, B.C. along the TCH. His map shows him the location of all gas stations along the way; the DeLorean gets 8km per liter regardless of how full the tank is ($12.5\frac{\ell}{100\,\mathrm{km}}$); the tank holds 50ℓ. Dr. Brown wants to stop at as few gas stations as possible.

(a) Model this problem as a subset selection problem.

(b) Create a greedy algorithm for the problem.

(c) Does your algorithm find a smallest possible subset of the available stations. Either prove that it does or give a counter-example that shows that it does not.

(Please do not use time travel to solve this problem.)

Q1 [40] We need to break a sequence of items of various weights into roughly equally weighted segments. The problem is represented as a sequence of n real weights $w = [w_0, w_1, .., w_{n-1}]$, a real number x, and an integer p. The goal is to break w into p segments such that no segment weighs more than x. For each segment, there is a penalty of the cube of the difference between the weight of the segment and x. If we represent the segments by an array of $p+1$ numbers $k[0]$, $k[1]$, ... , $k[p]$ where

$$0 = k[0] \leq k[1] \leq \cdots k[p-1] \leq k[p] = n,$$

then the segments are $[w_{k[0]}, w_1, .., w_{k[1]-1}]$, $[w_{k[1]}, w_{k[1]+1}, .., w_{k[2]-1}]$, ... and $[w_{k[p-1]}, w_{k[p-1]+1}, .., w_{k[p]-1}]$. An acceptable solution has

$$w_{k[i]} + w_{k[i]+1} + \cdots + w_{k[i+1]-1} \leq x$$

for each $i \in \{0, ..p\}$. An optimal solution minimizes the total penalty, which is

$$\sum_{i \in \{0,..p\}} \left(x - w_{k[i]} - w_{k[i]+1} - \cdots - w_{k[i+1]-1}\right)^3.$$

(a) Design a function to compute the cost (i.e., total penalty) of an optimal solution for inputs w, x, and p. Your function should return ∞ if there is no solution. Hint: For a given w, x, and p, you can define subproblems defined by integers i and q, such that $0 \leq i \leq n$ and $0 \leq q \leq p$; subproblem (i, q) is to find the cost of an optimal way to split the first i items of w into q segments, each of which weighs less than x. The original problem is just the subproblem such that $i = n$ and $q = p$.

(b) Convert your function from part (a) into a top-down dynamic programming algorithm.

(c) Convert your function from part (b) into a bottom-up dynamic programming algorithm.

(d) Adapt your solution from part (c) to find an optimal solution (i.e., an optimal k array).