Problem Set 4

CS157 - Spring 2009

Due: Tuesday April 7, 2009 10:30am

Instructions

For all of the problems we ask you to do the following:

- Give a recurrence relation before coming up with a dynamic programming algorithm. This is to isolate the subproblem which is first step in devising a dynamic algorithm. Do not worry about efficiency for the recurrence relation, but rather focus on isolating the subproblem. For example for the Fibonacci sequence the correct recurrence relation would look like:

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

- *Briefly* prove the correctness of your recurrence relation.
- Write up the pseudo-code for your algorithm. This involves defining a dynamic programming array by specifying what set of parameters index each array entry and the meaning of the content of the array entry and finally defining the order in which you will fill in the entries of the dynamic programming array.
- Analyze worst-case running time and storage requirements of your algorithm. Remember that there is a tradeoff for the power of dynamic programming, and that is it can have a relatively high polynomial run time.

Problem 1: Car Painting

Your job is to assist in painting cars. You are given a sequence of n cars $\{c_1, c_2..., c_n\}$ along with corresponding colors $\{p_1, p_2..., p_n\}$ to paint each car. By grouping the cars together by color you can avoid the penalty of having to get a different color paint before returning to your awesome job of car painting. So to minimize this penalty, before painting your manager lets you move each car at most 2 parking spots from where it currently is located. (Not more than 2 or else the customer won't be able to recognize their vehicle). That is, if a car is j-th in the inputted sequence it can be moved to the $j \pm 2$-th spot in the resulting sequence of freshly painted cars.

(a) Given as input a sequence of cars $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}$ with corresponding colors {teal, magenta, magenta, teal, teal, magenta, teal, magenta, teal, magenta}, what is the minimum number of color groups? What is resulting sequence of freshly painted magenta and teal cars?

(b) Give a recurrence relation for the Car Painting problem $CP(\{c_1, c_2..., c_n\}, \{p_1, p_2..., p_n\})$ which inputs the sequence of cars and the number of different colors and outputs the minimum number of color groups.

(c) Prove the correctness of your recurrence relation.

(d) Give a dynamic programming algorithm for a solution to the Car Painting problem that is polynomial in n. Analyze the worst-case running time and storage requirements of your algorithm.

Problem 2: Context-Free Grammar

A context-free grammar G is defined as $G = (V, \Sigma, R, S)$ where V is a finite set of non-terminals, Σ is a finite set of terminals, R is a relation from V to $(V \cup \Sigma)^*$, and S is the start variable. This defines a set of words which can be constructed by applying the rules. Note that each such word contains terminal symbols only. Consider the following context-free grammar G where $V = S, X, Y, Z$ $\Sigma = a, b, c$ S is the start symbol and R consists of the following production rules:

$$
\begin{aligned}
S &\rightarrow aX \\
S &\rightarrow bX \\
X &\rightarrow YY \\
X &\rightarrow aX \\
X &\rightarrow cZ \\
Y &\rightarrow bb \\
Z &\rightarrow c \\
Z &\rightarrow Xb
\end{aligned}
$$

Given a word w, the problem is to determine whether w is in the language generated by G.

(a) Given a word $w =$ aaacabbc, does it exist in the language generated by G?

(b) Give a recurrence relation for the Context-Free Grammar problem $CFG(w)$ which inputs a word w and outputs true if w is in the language generated by G (as defined in the example) and false otherwise.

(c) Prove the correctness of your recurrence relation.

(d) Give a dynamic programming algorithm for the Context-Free Grammar problem $CFG(w)$. Analyze the worst-case running time and storage requirements of your algorithm.

Problem 3: Job Scheduling

You are to build a scheduling machine that inputs n jobs $\{j_1, j_2..., j_n\}$, each with integer duration $d(j_i)$, an integer deadline $t(j_i)$ and a profit $p(j_i)$ and outputs a schedule of a **subset** of the jobs run in serial on a machine (where the first job starts at time 0). Not every job must be included in the schedule, but every included job must be completed by its deadline. The goal is to maximize the sum of the profits of jobs that are scheduled.

(a) Given as input 5 jobs with the corresponding functions defined over the domain $\{j_1, j_2, j_3, j_4, j_5\}$: $d(j_i) = \{2, 1, 8, 6, 50\}$, $t(j_i) = \{10, 5, 8, 15, 100\}$ and $p(j_i) = \{3, 9, 14, 1, 1\}$, what is the maximum profit that your scheduling machine can make? What is the schedule?

(b) Give a recurrence relation for the Job Scheduling problem $JS(\{j_1, j_2..., j_n\})$ which inputs the jobs and outputs the maximum profit. Assume that the duration, deadline and profit functions are defined for each of the jobs.

(c) Prove the correctness of your recurrence relation.

(d) Give a dynamic programming algorithm to find the maximum profit for the Job Scheduling problem. Analyze the worst-case running time and storage requirements of your algorithm.

Problem 4: Carpentry

A carpenter has a piece of wood of a certain length L that must be cut at positions $\{a_1, a_2..., a_n\}$ where a_i is the distance from the left end of the original piece of wood. Notice that after making the first cut, the carpenter now has two pieces of wood; after making the second cut, the carpenter has three pieces of wood, etc. Assume that the cost of making a cut in a piece of wood of length l is equal to l, and is the same no matter which position in that piece of wood is being cut. The goal of the carpenter is to minimize his costs while performing all of the cuts.

(a) Given a piece of wood of original length $L = 10$ and a set of positions to cut it at $\{a_1, a_2..., a_n\} = \{1, 4, 5, 8\}$, what is the sequence of cuts that minimizes the cost of all of the cuts?

(b) Give a recurrence relation for the Carpentry problem $C(L, \{a_1, a_2..., a_n\})$ which inputs the original length L and a set of cuts as defined above $\{a_1, a_2..., a_n\}$ and outputs the minimal cost in making all of the cuts.

(c) Prove the correctness of your recurrence relation.

(d) Give a top-down dynamic programming algorithm to find the minimum total cost for making all the cuts that minimizes the total cost. Analyze the worst-case running time and storage requirements of your algorithm.

(e) Give a bottom-up dynamic programming algorithm to find the minimum total cost for making all the cuts and the order in which to make the cuts that minimizes the total cost. Analyze the worst-case running time and storage requirements of your algorithm.