CSE 441T/541T Advanced Algorithms **Fall 2008**

Homework 2: Dynamic Programming

Assigned: September 17, 2008 *Due Date: October 1, 2008*

Whenever you are asked to give an algorithm for a problem, I expect you to do all of the following:

1. **give a clear, concise description of your algorithm;**
2. **prove that the algorithm outputs an optimal solution;**
3. **give the asymptotic time complexity of the algorithm;**
4. **prove that the time complexity is as claimed.**

You must submit your homework with a signed cover sheet attached to the front.

Note that the point values may not add up to 100; your score will be total points earned divided by total number of *required* points.

Core Problems

1. (10 points) Think back to the aliens problem in Homework 1. In that problem, we tried to minimize the total number of projects that were permitted to use the telescope, and each project made it more likely that the alien home world would be discovered. Now we suppose that the likelihood that a project will discover the alien home world depends on how many hours the project gets on the telescope. So (to you) the penalty of assigning a job is proportional to the length of the job.

   Additionally (!), Earth recently discovered parallel computing, so if two jobs overlap, *both* of them get to analyze the data from the telescope. Your goal is now to assign jobs to minimize the chance of discovering the home world.

   As in the previous problem, you are to assume that there is some project that covers each part of the interval [S,F), and that you must "avoid suspicion" by never allowing the telescope to be inactive. Each chosen project $p_i \in P$ will work the interval $[s_i, f_i)$ (precisely; no partial intervals are permitted, so projects may overlap). Give an efficient algorithm to select projects to cover the period $[S, F)$ that minimizes the total cost (i.e. total length of intervals) for all projects chosen, i.e.

$$\sum_{p_i \text{ chosen}} f_i - s_i.$$

2. (15 points) Professor Audubon has recorded the song of a bird in his backyard. The record consists of a time series Q of n successive pitch measurements $q_1 \ldots q_n$, taken once every 100 milliseconds. The professor wants to divide the song into its individual notes (intervals of constant pitch). Unfortunately, experimental noise randomly perturbs the pitches measured over the course of a single note, so that they do not remain constant.

To approximately divide the song into notes, the professor proposes the following approach. Let the *variance* of an interval $q_i \dots q_j$ of Q, denoted $\sigma^2(i,j)$, be defined in the usual statistical sense as

$$\sigma^2(i,j) = \sum_{k=i}^{j} \left(q_k - E(i,j)\right)^2 ,$$

where

$$E(i,j) = \frac{1}{j-i+1} \sum_{k=i}^{j} q_k$$

is the mean pitch of measurements $i..j$. If the measurement error in each pitch is small, then the variance of the interval corresponding to a single note should be low, while the variance over intervals combining distinct notes should be high. Hence, we want to divide Q into notes such that the sum of variances for all notes is is minimized.

There is one important problem with the above approach. The number of notes is unknown *a priori*. One could simply create a new note for each measurement, making the variance of each note (and hence the sum of variances for all notes) zero! To avoid this trivial solution, we penalize a proposed division into m notes by adding $m \cdot p$ to its cost, where $p > 0$ is a user-defined penalty.

To summarize, the goal is to divide the song Q into notes by selecting $m \leq n$ break points $b_1 \dots b_m$ (the ends of each note) so as to minimize

$$m \cdot p + \sum_{i=1}^{m+1} \sigma^2(b_{i-1}+1, b_i)$$

where $b_0 = 0$ and $b_{m+1} = n$.

Give an efficient algorithm for this problem. Your solution should run in time $O(n^2)$. *Hint*: show how to compute $\sigma^2(i,j)$ in constant time for any (i,j).

3. (10 pts) Think back to the skis and skiers problem of Homework 1. Recall that the goal is to match each skier height h_i to a ski length l_i so as to minimize the sum of absolute differences $\sum_i |h_i - l_i|$.

   Suppose that the numbers of skiers and skis are n and $m > n$ respectively. As before, we may not match more than one pair skis to the same skier; however, in this version of the problem, not all pairs of skis must be used. Given an efficient algorithm to assign a pair of skis to each skier. Be sure to analyze your time complexity as a function of both m and n.

   (**Note**: you may want to look at the solution to the $m = n$ case in Homework 1 before attempting this problem! You need not re-prove anything about that algorithm on this homework.)

4. (15 points) Consider the problem of word-wrapping a paragraph. A paragraph is an ordered list of n words, where word w_i is ℓ_i letters long. You want to divide the paragraph into a sequence of lines, each containing at most L letters. (No word is more than L letters long.)

   Suppose a line contains words $w_i \dots w_j$. The total length $W(i,j)$ of this line is defined by

$$W(i,j) = j - i + \sum_{k=i}^{j} \ell_k .$$

This length accounts for a single space between successive pairs of words on the line. The *slop* $S(i, j)$ of this line is defined to be $L - W(i, j)$, the total number of unused spaces at the end of the line. Note that in any feasible solution, the slop of each line must be non-negative. (The cubed slop criterion is a simplified version of what is actually used in, e.g., TeX for paragraph wrapping.)

Just to make things concrete, consider the example paragraph "Now is the time for all good men.", and suppose $L = 10$. One feasible solution is

```
Now is the
time for
all good
men.
```

This solution has four lines of lengths 10, 8, 8, and 4; the corresponding slops are 0, 2, 2, and 6.

Your goal is to find a division of the input paragraph into lines that minimizes the sum, over all lines *except the last*, of the *cubed* slop of each line. (We omit the last line because it can in general be much shorter than the others.) For example, the total cost of the above solution is $0^2 + 2^3 + 2^3 = 16$.

Give an efficient algorithm for this problem.

Advanced Problem

This problem is *required* only for CSE 541 students. 441 students may receive extra credit for a correct solution.

5. (15 points) The *Euclidean traveling-salesman problem* requires one to determine the shortest closed tour that connects a given set of n points in the plane. Unlike a "path", a tour must start and finish at the same point. A farcical look at Euclidean Traveling Salesman problems is at:

   `http://www.oberlin.edu/math/faculty/bosch/making-tspart-page.html`

   In this problem we consider a simpler problem: *two-way tours*, which start at the leftmost point, go strictly left to right to the rightmost point, and then go strictly right to left back to the starting point. An example of such a tour is shown on the left of:

   `http://www.cs.wustl.edu/~pless/546/lectures/f7_1.gif`

   Describe an $O(n^2)$-time dynamic programming algorithm for determining an optimal two-way tour. You may assume that no two points have the same x coordinate.

   Hint: Scan left to right, maintaining optimal possibilities for the two parts of the tour.