CMPS 102 Homework 6, Spring ’04
4 problem, 25 points, due Tuesday May 17th

1. (5 pts) Consider the matrix chain multiplication problem with 6 matrices \(A_1 \cdot A_2 \cdots A_6\) and dimension sequence: 5, 10, 3, 12, 5, 50, 6. Thus \(A_1\) has 5 rows and 10 columns, and \(A_6\) has 50 rows and 6 columns. Compute the table of \(m(i, j)\) values giving the cost of the best way to multiply the \(A_i\) to \(A_j\) subchain (as defined by equation 15.12). Also give an optimal parenthesization for multiplying these 6 matrices.

2. (10 pts) Assume you are planning a canoe trip down a river. The river has \(n\) trading posts numbered 1 to \(n\) going downstream. You will start your trip at trading post number 1 and end at trading post number \(n\). Let \(R(i, j)\) be the cost of renting a canoe at trading post \(i\) and returning it at trading post \(j\), where \(j > i\). Assume that you always want to go down river, so the costs if \(j \leq i\) are irrelevant. Find the cheapest sequence of rentals that allow you to complete your trip. Aim for an algorithm running in \(O(n^2)\) time.

For example, if \(n = 4\) and the costs are:

\[
\begin{array}{c|cccc}
   & 2 & 3 & 4 \\
\hline
   1 & 15 & 25 & 35 \\
   2 & -- & 12 & 16 \\
   3 & -- & -- & 5 \\
\end{array}
\]

then the cheapest sequence of canoe rentals to travel the river would be to rent from 1 to 3, and then from 3 to 4 for a cost of 25 + 5 = 30.

On the other hand, if the costs were:

\[
\begin{array}{c|cccc}
   & 2 & 3 & 4 \\
\hline
   1 & 20 & 15 & 30 \\
   2 & -- & 5 & 10 \\
   3 & -- & -- & 20 \\
\end{array}
\]

then taking one canoe all the way from 1 to 4 and renting 1 to 2 and then 2 to 4 are the cheapest solutions (both cost 30). (Note that renting from 1 to 3 is cheaper than going 1 to 2, but the 1 to 3 rental is not in any of the cheapest 1 to 4 solutions.)

Let \(C(k)\) be the cost of the cheapest sequence of canoe rentals starting from trading post 1 and returning the last canoe rented at trading post \(k\).

First, assume an optimal sequence of rentals changes canoes at some trading post \(j\). What subproblems are also solved optimally by (parts of) this rental sequence? Prove your answer.

Second, derive a recurrence for \(C(k)\) in terms of \(C(j)\) values where \(j < k\).
Third (turn in only if you do not get the final part), use your recurrence to construct
a recursive algorithm for computing the $C(k)$ values.

Fourth, give a bottom-up iterative algorithm for computing the $C(j)$ values.

Finally, show how keeping a little additional information allows the a cheapest sequence
of canoe rentals to be printed out.

3. (10 pts) Coins and Change. Assume that you are working a cash register in a country
that has strange values for its coins (for example: 1, 5, 8, and 10) and you have
to give some amount (like 16) in change using the fewest possible coins. For sensible
systems, the greedy method works (take as many of the largest coin as possible without
exceeding the amount, go the next largest coin, etc.). However, with this example it
is better to take two coins of value 8 than a 10, 5, and 1. For this problem you will
develop a dynamic programming algorithm for the Coins and Change problem.

Formally, the Coins and Change problem is:

Given $n$, a list of $k$ integer coin values $v_1 = 1 < v_2 < v_3 < \ldots < v_k$ and an
amount $a$, find a way to give value exactly $a$ using the fewest possible coins.
Assume that you have plenty of coins of each denomination.

Thus a potential solution is vector $N = (n_1, n_2, \ldots, n_k)$ where each $n_i$ is non-negative
and indicates how many value $v_i$ coins to give. The solution $N$ is feasible if $\sum_{i=1}^{k} n_i v_i = a$. The solution $N$ is optimal if $N$ is feasible and for every feasible solution $N' = (n'_1, \ldots, n'_k)$ the $\sum_{i=1}^{k} n_i \leq \sum_{i=1}^{k} n'_i$.

Before proceeding, make sure you understand the problem. Note that the requirement
that $v_1 = 1$ ensures that it is possible to give change for any integer amount.

First, assume we have an optimal solution $N$. Find a first or last choice made by the
solution and use this first or last choice to identify other (smaller) Coins and Change
problems that optimally solved by (part of) the solution $N$. Prove that these smaller
problems are indeed solved optimally by (parts of) the optimal solution.

Second, focus simply on the number of coins in optimal solutions. Use the insight from
the first part to derive a recurrence for the number of coins in optimal solutions.

Third, sketch a top-down recursive algorithm to compute the number of coins in opti-
mal solutions.

Fourth, describe what kind of table can be used to memoize your recursion so that
repeated subproblems do not need to be recomputed.

Next, give pseudo-code for an iterative bottom-up algorithm that fills in the table.

Finally, indicate how storing some additional information can allow an optimal solution
(the vector $N$) to be constructed.