1. (15 pts) Suppose there are $n$ trading posts along a river. At any of the posts, you can rent a canoe to be returned at any other post downstream. (It is impossible to paddle against the current.) For each possible departure point $i$ and each possible arrival point $j$, the cost of taking a rental from $i$ to $j$ is known. However, it can happen that the cost of renting from $i$ to $j$ is higher than the total cost of a series of shorter rentals. In this case, you can return the first canoe at some post $k$ between $i$ and $j$ and continue your journey in a second canoe. There is no extra charge for changing canoes this way. Give an efficient algorithm to determine the minimum cost of a trip by canoe from each possible departure point $i$ to each possible arrival point $j$. In terms of $n$, how much time is needed by your algorithm?

2. (15 pts) Recall the Knapsack problem in which you are given $n$ items, each item $i$ with a nonnegative weight $w_i$ and a distinct value $v_i$. You are interested in finding the subset $S$ of maximum value $\sum_{i \in S} v_i$ whose weight $\sum_{i \in S} w_i$ does not exceed budget $W$. In class, we saw how to find the maximum value achievable by any subset. Design an $O(nW)$-time algorithm to construct the set $S$ of maximum value.

3. (15 pts) Suppose you have $n$ boys and $n$ girls, each of whom specifies a set of $K$ friends of the opposite gender with whom they are willing to attend the high school dance. Prove that it is always possible to find a matching between boys and girls such that everyone goes with one of their friends. (In this problem, one-sided friendships do not exist.)

4. (15 pts) Given a string of characters $c_1 \ldots c_n$, we say that a substring $c_i \ldots c_j$ for $1 \leq i \leq j \leq n$ is a palindrome if it reads the same forward and backwards. For example, “abacaba” is a palindrome. Give an $O(n^2)$-time algorithm to find the longest palindrome substring in the input string $c_1 \ldots c_n$.

5. (20 pts) You are given a rectangular piece of cloth with dimensions $X \times Y$, where $X$ and $Y$ are positive integers, and a list of $n$ types of products that can be made using the cloth. For each product type $i$, you know that a rectangle of cloth of dimensions $a_i \times b_i$ is needed and that the selling price of the product is $c_i$. Assume that $a_i$, $b_i$ and $c_i$ are all positive integers. You have a machine that can cut any rectangular piece of cloth into two pieces either horizontally or vertically.

Design an algorithm that determines the best return on the $X \times Y$ piece of cloth, that is, a strategy for cutting the cloth so that the products made from the resulting pieces give the maximum sum of selling prices. You are free to make as many copies
of a given product as you wish, or none, if desired. You may find the following hints helpful:

(a) First, prove that w.l.o.g., cuts are made only on integer boundaries.
(b) Develop a recurrence for the maximum return that can be obtained from a cloth of arbitrary integral dimension $i \times j$. (The natural thought would be: we either use the cloth for a single copy of a product, or . . . )
(c) Use this recurrence to design the algorithm.