CS 325
Homework 3

Problem 1 (3 points)

Rod Cutting: Show, by means of a counterexample, that the following "greedy" strategy does not always determine an optimal way to cut rods. Define the density of a rod of length i to be p_i / i, that is, its value per inch. The greedy strategy for a rod of length n cuts off a first piece of length i, where $1 \leq i \leq n$, having maximum density. It then continues by applying the greedy strategy to the remaining piece of length $n - i$.

Problem 2 (3 points)

Modified Rod Cutting: Consider a modification of the rod-cutting problem in which, in addition to a price p_i for each rod, each cut incurs a fixed cost of c. The revenue associated with a solution is now the sum of the prices of the pieces minus the costs of making the cuts. Give a dynamic-programming algorithm (pseudocode) to solve this modified problem.

Problem 3 (6 points)

Making Change: Given coins of denominations (value) $1 = v_1 < v_2 < \ldots < v_n$, we wish to make change for an amount A using as few coins as possible. Assume that v_i's and A are integers. Since $v_1 = 1$ there will always be a solution. Formally, an algorithm for this problem should take as input an array V where V[i] is the value of the coin of the ith denomination and a value A which is the amount of change we are asked to make. The algorithm should return an array C where C[i] is the number of coins of value V[i] to return as change and m the minimum number of coins it took. You must return exact change so $\sum_{i=1}^{n} \mathrm{V[i]} \cdot \mathrm{C[i]} = \mathrm{A}$

The objective is to minimize the number of coins returned or: $m = min \sum_{i=1}^{n} \mathrm{C[i]}$

(a) Describe and give pseudocode for a dynamic programming algorithm to find the minimum number of coins needed to make change for A.

(b) What is the theoretical running time of your algorithm?

Problem 4 (18 points)

Acme Super Store is having a contest to give away shopping sprees to lucky families. If a family wins a shopping spree each person in the family can take any items in the store that he or she can carry out, however each person can only take one of each type of item. For example, one family member can take one television, one watch and one toaster, while another family member can take one television, one camera and one pair of shoes. Each item has a price (in dollars) and a weight (in pounds) and each person in the family has a limit in the total weight they can carry. Two people cannot work together to carry an item. Your job is to help the families select items for each person to carry to maximize the total price of all items the family takes.

(a) Write an efficient algorithm (verbal description and pseudo-code) to determine the maximum total price of items for each family and the items that each family member should select.

(b) What is the theoretical running time of your algorithm for one test case given N items, a family of size F, and family members who can carry at most M_i pounds for $1 \leq i \leq F$.

(c) Implement your algorithm by writing a program named "shopping". The program should satisfy the specifications below.

Input: The input file named "shopping.txt" consists of T test cases

- T ($1 \leq T \leq 100$) is given on the first line of the input file.
- Each test case begins with a line containing a single integer number N that indicates the number of items ($1 \leq N \leq 100$) in that test case
- Followed by N lines, each containing two integers: P and W. The first integer ($1 \leq P \leq 5000$) corresponds to the price of object and the second integer ($1 \leq W \leq 100$) corresponds to the weight of object.
- The next line contains one integer ($1 \leq F \leq 30$) which is the number of people in that family.
- The next F lines contains the maximum weight ($1 \leq M \leq 200$) that can be carried by the ith person in the family ($1 \leq i \leq F$).

Output: Written to a file named "results.txt". For each test case your program should output the maximum total price of all goods that the family can carry out during their shopping spree and for each the family member, numbered $1 \leq i \leq F$, list the item numbers $1 \leq N \leq 100$ that they should select. List both family members and the corresponding items in order.

Sample Input (*comments* are not part of the file, they are added for clarification)

2	*// 2 test cases*
3	*// 3 items for test case 1*
72 17	*// P and W of item 1 (case 1)*
44 23	*// P and W of item 2 (case 1)*
31 24	*// P and W of item 3 (case 1)*
1	*// 1 family member (case 1)*
26	*// max weight for family member 1 (case 1)*
6	*// 3 items for test case 2*
64 26	*// P and W of item 1 (case 2)*
85 22	
52 4	
99 18	
39 13	
54 9	
4	
23	
20	
20	
36	

Sample Output (in some cases, there are multiple solutions, providing only one of these solutions is enough)

Test Case 1

Total Price 72

Member Items

1: 1

Test Case 2

Total Price 568

Member Items

1: 3 4

2: 3 6

3: 3 6

4: 3 4 6