CSci 231 Homework 8 [*]

Dynamic Programming and Greedy Algorithms

CLRS Chapter 15 and 16

Use a (single) separate sheet of paper for each problem. Be concise.

1. A game-board consists of a row of n fields, each consisting of two numbers. The first number can be any positive integer, while the second is 1, 2, or 3. An example of a board with $n = 6$ could be the following:

17	2	100	87	33	14
1	2	3	1	1	1

   The object of the game is to jump from the first to the last field in the row. The top number of a field is the cost of visiting that field. The bottom number is the maximal number of fields one is allowed to jump to the right from the field. The cost of a game is the sum of the costs of the visited fields.

   Let the board be represented in a two-dimensional array $B[n, 2]$. The following recursive procedure (when called with argument 1) computes the cost of the cheapest game:

```
Cheap(i)

    IF i>n THEN return 0
    x=B[i,1]+Cheap(i+1)
    y=B[i,1]+Cheap(i+2)
    z=B[i,1]+Cheap(i+3)
    IF B[i,2]=1 THEN return x
    IF B[i,2]=2 THEN return min(x,y)
    IF B[i,2]=3 THEN return min(x,y,z)

END Cheap
```

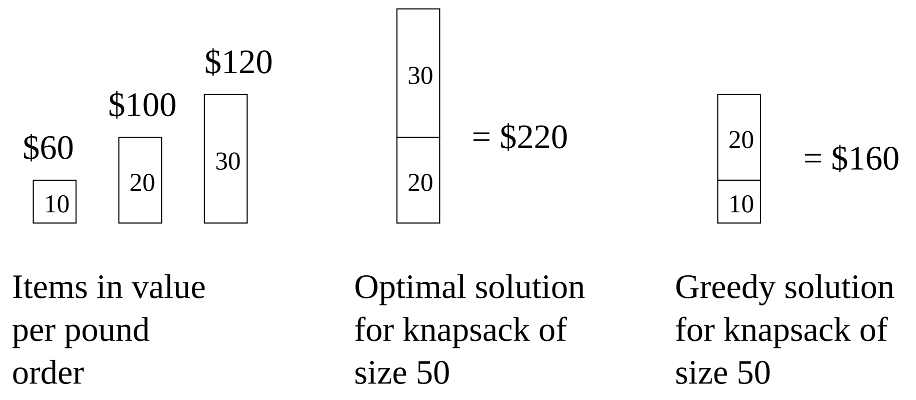
[*]Collaboration is allowed and encouraged, if it is constructive and helps you study better. Remember, exams will be individual. List the names of the collaborators.

(a) Analyze the asymptotic running time of the procedure.

(b) Describe and analyze a more efficient algorithm for finding the cheapest game.

2. In this problem we consider the 0-1 KNAPSACK PROBLEM: Given n items, with item i being worth $v[i]$ dollars and having weight $w[i]$ pounds, fill a knapsack of capacity m pounds with the maximal possible value.

Example: Given a knapsack of capacity 50, the maximal value obtainable with three items of value \$60, \$100, and \$120 and weights 10, 20, and 30, respectively, is \$220.



$60
10
$100
20
$120
30
Items in value per pound order
30
20
= $220
Optimal solution for knapsack of size 50
20
10
= $160
Greedy solution for knapsack of size 50

The algorithm `Knapsack(i,j)` below returns the maximal value obtainable when filling a knapsack of capacity j using items among items 1 through i (`Knapsack(n,m)` solves our problem). The algorithm works by recursively computing the best solution obtainable *with* the last item and the best solution obtainable *without* the last item, and returning the best of them.

```
Knapsack(i,j)

  IF w[i] <= j THEN
    with = v[i] + Knapsack(i-1, j-w[i])
  ELSE
    with = 0
  END IF
  without = Knapsack(i-1,j)
  RETURN max{with, without}

END Knapsack
```

(a) Show that the running time T of `Knapsack`(n, m) is exponential in n or m. (*Hint:* look at the case where $w[i] = 1$ for all $1 \leq i \leq n$ and show that $T(n, m) = \Omega(2^{\min(m,n)})$).

(b) Describe an $O(n \cdot m)$ algorithm for computing the value of the optimal solution.

3. Imagine now that the items in the problem above are such that you can take fractions of items. With this modified condition, the problem is called the FRACTIONAL KNAPSACK PROBLEM. Identify a greedy strategy to solve the problem. Prove that this strategy correctly identifies an optimal solution for all possible inputs.

   Analyse the running time of your algorithm and compare it with the 0-1 KNAPSACK PROBLEM running time.

   Does it pay off being greedy (in this case)?

4. Suppose you are in charge of planning a party for Bowdoin College. The college has a hierarchical structure, which forms a tree rooted at President Mills. On the very last level are the faculty, grouped by department. (I have such a chart in my office, if you need to be visual; see attached). Each faculty has "underneath" all students taking a class with him/her that particular semester. Assume that every person is listed at the highest possible position in the tree and there are no double affiliations (everybody has one and only one supervisor in this hierarchy and no student is in more than one class).

   You have access to a secret database which ranks each faculty/staff/student with a conviviality rating (a real number, which can be negative if the person is really difficult or boring). In order to make the party fun for everybody, President Mills does not want both a faculty/staff/student and his or her immediate "supervisor" to attend.

   You are given a tree that describes the strucure of Bowdoin College. Each node has a (down) pointer to its left-most child, and a (right) pointer to its next sibling (if unclear read Section 10.4 in CLR). Each node also holds a name and a conviviality ranking. Describe an algorithm to make up a guest list that maximizes the sum of the conviviality rankings of the guests. Analyze the running time of your algorithm.