

CSE 101 Homework 8
Dynamic Programming
Due Thursday, June 6

Gizmos (20 points) Consider the following problem. You wish to purchase (at least) n identical gizmos. Gizmos come in packages of different sizes and different prices. You can buy any number of packages of each size, as long as the total number is at least n . You wish to find the minimum total price of such a set of packages.

The input is given as n and an array $Packages[1..m]$, where each $Package[i]$ has a positive integer field $Package[i].size$ and a positive real field $Package[i].price$ giving the number of gizmos in the package and the price of the package.

A recursive algorithm to solve this problem is:

$BestPrice[n : positiveinteger, Packages[1..m] : \text{array of pairs (size: integer, price: real).}]$

1. $MinPrice \leftarrow \text{inf};$
2. For $d = 1$ to m do:
3. begin;
4. IF $Packages[d].size \geq n$ THEN $TempPrice \leftarrow Packages[d].price$
5. ELSE $TempPrice \leftarrow Packages[d].price + BestPrice(n - Packages[d].size, Packages);$
6. IF $TempPrice < MinPrice$ THEN $MinPrice \leftarrow TempPrice;$
7. end;
8. Return $MinPrice$.

Part 1: 2 points Show the recursion tree of the above algorithm on the following input: $n = 6$, packages: buy 5 for \$ 12, 3 for \$8 or 2 for \$6.

Part 2: 3 points Give a bound on the worst-case number of recursive calls the recursive algorithm could make in terms of n and m .

Part 3: 10 points Give a dynamic programming version of the recurrence.

Part 4: 3 points Give a time analysis of this dynamic programming algorithm, in terms of n and m .

Part 5: 2 points Show the array that your algorithm produces on the above example.

For each of the following three problems, describe the fastest dynamic programming algorithm you can find, and give a time analysis (in terms on any of the given parameters).

Non-consecutive sums Given an array of integers, $A[1..n]$, find the subset S of positions so that no two consecutive numbers i and $i + 1$ are both in S , which maximizes $\sum_{i \in S} A[i]$.

Interleaving Consider two binary strings $x_1, \dots, x_n$ and $y_1, \dots, y_m$. An *interleaving* of x and y is a string $z_1 \dots z_{n+m}$ so that the bit positions of z can be partitioned into two disjoint sets X and Y , so that looking only at the positions in X , the sub-sequence of z produced is x and looking only at the positions of Y , the sub-sequence is y . For example, if $x = 1010$ and $y = 0011$, $z = 10001101$ is an interleaving because the odd positions of z form x , and the even positions form y . The problem is: given x, y and z , determine whether z is an interleaving of x and y .

Library storage-20pts A library has n books that must be stored in alphabetical order on adjustable height shelves. Each book has a height and a thickness. The width of the shelf is fixed at W , and the sum of the thicknesses of books on a single shelf must be at most W . The next shelf will be placed on top, at a height equal to the maximum height of a book in the shelf. Give an algorithm that minimizes the total height of shelves used to store all the books. You are given the list of books in alphabetical order, $b_i = (h_i, t_i)$, where h_i is the height and t_i is the thickness, and must organize the books in that order.