

CS 170 HW 7

Due **2021-3-9, at 10:00 pm**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, write “none”.

DP solution writing guidelines:

Try to follow the following 3-part template when writing your solutions.

- Define a function $f(\cdot)$ in words, including how many parameters are and what they mean, and tell us what inputs you feed into f to get the answer to your problem.
- Write the “base cases” along with a recurrence relation for f .
- Prove that the recurrence correctly solves the problem.
- Analyze the runtime and space complexity of your final DP algorithm.

2 Counting Targets

We call a sequence of n integers x_1, \dots, x_n *valid* if each x_i is in $\{1, \dots, m\}$.

- (a) Give a dynamic programming-based algorithm that takes in n, m and “target” T as input and outputs the number of distinct valid sequences such that $x_1 + \dots + x_n = T$. Your algorithm should run in time $O(m^2n^2)$.
- (b) Give an algorithm for the problem in part (a) that runs in time $O(mn^2)$.
Hint: let $f(s, i)$ denotes the number of length- i valid sequences with sum equal to s . Consider defining the function $g(s, i) := \sum_{t=1}^s f(t, i)$.

3 Nightmare

Give an algorithm to find the number of ways you can place knights on an N by M ($M < N$) chessboard such that no two knights can attack each other (there can be any number of knights on the board, including zero knights). Clearly describe your algorithm and prove its correctness. The runtime should be $O(2^{3M} M \cdot N)$.

(Please provide a 3-part solution)

4 Geometric Knapsack

Suppose we have a rectangular paper of side lengths X, Y , where X, Y are positive integers, and a set of n products that can be made of the paper. Each product is a rectangle of dimensions $a_i \times b_i$ and of value c_i , where all these numbers are positive integers. Suppose we can only cut the paper horizontally and vertically.

Describe and analyze an algorithm that determines the maximum value of the products that can be made out of the single $X \times Y$ paper. You may produce a product multiple times, or not at all if you wish.

5 GCD annihilation

Let x_1, \dots, x_n be a list of positive integers given to us as input. We repeat the following procedure until there are only two elements left in the list:

Choose an element x_i in $\{x_2, \dots, x_{n-1}\}$ and delete it from the list at a cost equal to the greatest common divisor of the undeleted left and right neighbors of x_i .

We wish to make our choices in the above procedure so that the total cost incurred is minimized. Give a $\text{poly}(n)$ -time dynamic programming-based algorithm that takes in the list x_1, \dots, x_n as input and produces the value of the minimum possible cost as output. You may assume that we are given an $n \times n$ sized array where the i, j entry contains the GCD of x_i and x_j , i.e., you may assume you have constant time access to the GCDs.