CSCI B505 Fall 19: Programming assignment 3

Due online: October 18, 11:59pm EST

Submit your work online using Canvas (no assignments will be accepted on paper this time). You can use LaTeX, Word, or even pen and paper for the write-up. But please try to submit a PDF file. Use the lab sessions prior to the due date to get help from your AI/UI, if necessary.

What to submit

For this assignment you will be submitting two things:

1. Source code. Please, follow good coding practices: use indentation, write comments, etc.
2. Write-up. This should contain:
   - Description of how you solved the problem, including the recursive formulation, base cases, whether you are using top-down or bottom-up and why.
   - Justifications for any choices you've made.
   - Conclusions and analysis of the results.

Solve **one of the two problems** below using dynamic programming. **Each problem is worth 100 points, but you can only submit one**. It's up to you which two problems to choose. You can use either top-down dynamic programming (memoization) or bottom-up.

Problem 1

The input for this problem consists of two integers n and k, where $1 \leq k \leq 10$ and a sequence of distinct numbers $c_1 < c_2 < \ldots < c_k$, where each c_i is an integer between 1 and 10. Your program needs to output the number of ways one can write n as a sum of integers from the set $\{c_1, c_2, \ldots, c_k\}$. You only have to output the number of different partitions, not the partitions themselves. Furthermore, you can output either (the choice is up to you):

- The number itself (note, that it can be very large).
- The last digit of this number (note that in this case it suffices to always take the recurrence mod 10 in each step).

Run your program for the following five inputs:

1. $n = 10^3$, $k = 2$, $c_1 = 1, c_2 = 2$.
2. $n = 10^3$, $k = 3$, $c_1 = 1, c_2 = 3, c_3 = 5$.
3. $n = 2 \cdot 10^3$, $k = 3$, $c_1 = 2, c_2 = 3, c_3 = 5$.
4. $n = 10^3$, $k = 10$, $c_1 = 1, c_2 = 2, c_3 = 3, c_4 = 4, c_5 = 5, c_6 = 6, c_7 = 7, c_8 = 8, c_9 = 9, c_{10} = 10$.
5. $n = 2 \cdot 10^3$, $k = 4$, $c_1 = 2$, $c_2 = 4$, $c_3 = 5$, $c_4 = 6$.

Example: **Input:** $n = 5, k = 3, c_1 = 1, c_2 = 3, c_3 = 4$.

Output: 6.

Partitions:

1. $1 + 1 + 1 + 1$
2. $1 + 1 + 3$
3. $1 + 3 + 1$
4. $3 + 1 + 1$
5. $4 + 1$
6. $1 + 4$

Problem 2

You are given a set of jobs. Each job i has a start time s_i and a finish time f_i. It also has a weight w_i. Two jobs are compatible if their start-finish intervals don't overlap. The goal is to find the maximum weight subset of mutually compatible jobs (a subset with the maximum sum of weights).

Run your program on the two inputs provided. You have to output only the maximum weight, not the actual subset. Each input file contains multiple lines, where each line contains three numbers describing a job: s_i, f_i, w_i. Each of these numbers is an integer between 1 and 200. See example below:

Input: 1 2 50
3 5 30
6 19 100
2 100 200

Output: 250