University of Washington
Department of Computer Science and Engineering
CSE 417, Autumn 2020

November 18, 2020

Homework 7, Due Wednesday, November 25, 2020

On all problems provide justification of your answers. Provide a clear explanation of why your algorithm solves the problem, as well as a justification of the run time. Since this assignment is from the dynamic programming section - your algorithms should use dynamic programming!

Problem 1 (10 points) Weighted Independent Set on a Path:

The weighted independent set problem is: Given an undirected graph $G = (V, E)$ with weights on the vertices, find an independent set of maximum weight. A set of vertices I is independent if there are no edges between vertices in I. This problem is known to be NP-Complete.

For a simpler problem, consider a graph that is a path, where the vertices are $v_1, v_2, \ldots, v_n$, with edges between v_i and v_{i+1}. Suppose that each node v_i has an associated weight w_i. Give an algorithm that takes an n vertex path with weights and returns an independent set of maximum total weight. The run time of the algorithm should be polynomial in n.

Problem 2 (10 points) Task Choice:

Suppose that each week you have the choice of a high stress task, a low stress task, or no task. If you take a high stress task in week i, you are not allowed to take any task in week $i+1$. For n weeks, the high stress tasks have payoff $h_1, \ldots, h_n$, and the low stress tasks have payoff $l_1, \ldots, l_n$, and not doing a task has payoff 0. Give an algorithm which given the two lists of payoffs, maximizes the value of tasks that are performed over n weeks. The run time of the algorithm should be polynomial in n.

Problem 3 (10 points) Strict Subset Sum:

The strict subset sum problem is: Given a set of values $\{s_1, \ldots, s_n\}$, and an integer K, is there a subset of the items that sum to exactly K. Design an algorithm that solves the strict subset sum, and finds a set that sums to K with as large a number of items as possible. Your algorithm should have runtime polynomial in n and K.

Problem 4 (10 points) Counting solutions to the subset sum:

The subset sum counting problem is: Given a set of values $S = \{s_1, \ldots, s_n\}$, and an integer K, determine the number of subsets of S that sum to exactly K. Design an algorithm that solves the subset sum counting problem. Your algorithm should have runtime $O(nK)$.

Problem 5 (20 points) Programming: Electoral College Ties:

Determine how many different ways the Electoral College can result in a 269-269 tie in the 2020 US Presidential election.

How the electoral college works: Each US state plus the District of Columbia has a given number of delegates based on its population. An election is held in each state and the winner of that election receives all of the delegates for that state. The person receiving the largest number of delegates is then the president of the US. (This method has the possibility that the person elected president is not necessarily the person winning the most votes nationally.)

For this problem, you are given a list of the number of votes each state has in the electoral college, and you are asked to compute the number of ways that these votes can be allocated to reach a 269-269 tie. We are assuming that there are only two candidates, and that states allocate all of their votes to one candidate or the other.

Obviously, use dynamic programming. There are lots of different ways of reaching a tie - so many that you will need to use 64-bit integers (e.g., long ints).

The data is available here so you can copy the arrays into your program.

a.) How many ways are there for the electoral college to result in a 269-269 tie.

b.) Find a group of states that can reach exactly 269 votes.

c.) Provide your algorithmic code.

d.) What is the runtime of your algorithm (as a function of the number of states, and of the number of electoral votes). Justify your answer.