University of Washington
Department of Computer Science and Engineering
CSE 417, Autumn 2023

November 20, 2023

Homework 8, Due Wednesday, November 29, 2023

On problems one through three, 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. You will want to do problems 2 and 3 before tackling the programming problems.

Problem 1 (10 points) Word segmentation:

(This problem is based on problem 5 on Page 316 of the text without the excessive verbiage.) The word segmentation problem is: given a string of characters $Y = y_1 y_2 \dots y_n$, optimally divide the string into consecutive characters that form words. (The motivation is that you are given a text string without spaces and have to figure out what the words are. For example, *"meetateight"* could be *"meet ate ight"*, *"me et at eight"* or *"meet at eight"*.) The problem is to find the best possible segmentation. We assume we have a function *Quality* which returns an integer value of the goodness of a word, with strings that correspond to words getting a high score and strings that do not correspond to words getting a low score. The overall quality of a segmentation is the sum of the qualities of the individual words.

Give a dynamic programming algorithm to compute the optimal segmentation of a string. You can assume that calls to the function *Quality* take constant time and return an integer value. What is the runtime of your algorithm?

Problem 2 (10 points) Counting solutions to the subset sum:

The subset sum counting problem is: Given a set of values $S = \{s_1, \dots, s_n\}$, 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 3 (10 points) Smallest subset sum:

The smallest subset sum problem is: Given a set of values $S = \{s_1, \dots, s_n\}$, and an integer K, find a subset of S that sums to exactly K with the minimum number of items. (If there is no subset that sums to K, then the answer is undefined.) Design an algorithm that solves the smallest subset sum problem. Your algorithm should have runtime $O(nK)$ and use space $O(nK)$. (Update: the space complexity bound was changed to $O(nK)$. While $O(n + K)$ is possible, I don't want that to be the focus of the problem.)

Programming Problem 4 (10 points) Number of Electoral College Ties:

Determine how many different ways the Electoral College could result in a 269-269 tie in the 2024 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 were only two candidates, and that states allocated 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). In Java or C#, you need to declare these with type **long**.

The data is available here so you can copy the arrays into your program. Use the 2024 data (but you can experiment with other years as well.) The data goes back to 1964, which is the year that the number of votes became 538 (prior to that there were an odd number of votes.) The allocation of electoral votes is changed once a decade based on the census.

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

b.) Provide your algorithmic code.

c.) 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.

Debugging hints. You should first try your code on a small data set (e.g., with four states) that you can work out the answer by hand. For the full data set, there are some vote counts, where you can determine how many times they are reached by looking at the data (such as the number of ways of getting exactly 3 votes, and the number of ways of getting exactly 538 votes.)

Programming Problem 5 (10 points) Number of Electoral College Ties:

Determine the fewest states that a candidate could win to reach exactly 269 electoral votes, leading the to a 269-269 tie in the Electoral College in the 2024 US Presidential election.

Use the data set for problem 4 for this problem. It is probably easiest to write a new dynamic program for this problem, instead of extracting the data from your previous program (although the structure is naturally similar.)

a.) What is the smallest number of states that sum to 269 electoral votes?

b.) Find a smallest set of states that sum to 269 electoral votes. (Give the names of the states - note that the dataset gives you an array of state names.)

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.