CSE 321 Homework 5

Due date: 09 / 01 / 2020

1-) Suppose that you are running a lightweight consulting business - just you, two associates, and some rented equipment. Your clients are distributed between the East Coast and the West Coast of USA, and this leads to the following question. Each month, you can either run your business from an office in New York (NY) or from an office in San Francisco (SF). In month i, you will incur an operating cost of N_i if you run the business out of NY; you will incur an operating cost of S_i if you run the business out of SF. (It depends on the distribution of client demands for that month.) However, if you run the business out of one city in month i, and then out of the other city in month i + 1, then you will incur a fixed moving cost of M to switch base offices.

Given a sequence of n months, a plan is a sequence of n locations - each one equal to either NY or SF - such that the i^{th} location indicates the city in which you will be based in the i^{th} month. The cost of a plan is the sum of the operating costs for each of the n months, plus a moving cost of M for each time you switch cities. The plan can begin in either city.

The problem: Given a value for the moving cost M and sequences of operating costs N_1 ..... N_n and S_1 ..... S_n, find a plan of minimum cost. (Such a plan will be called optimal.)

For example, suppose that n=4, M=10 and the operating costs are given by the following table:

#	Month 1	Month 2	Month 3	Month 4
NY	1	3	20	30
SF	50	20	2	4

Then the plan of minimum cost would be the sequence of locations:

[NY, NY, SF, SF]

with a total cost of 1 + 3 + 2 + 4 + 10 = 20, where the final term of 10 arises because you change locations once.

Design a dynamic programming algorithm that takes values for n, M, and sequences of operating costs N_1 ..... N_n and S_l .....S_n, and returns the cost of an optimal plan. Implement your algorithm and analyze its worst-case running time. Explain your algorithm in your report file.

2-) Suppose that you attend a symposium which has many simultaneous sessions. The lengths of the sessions are not fixed and they begin at various times. As a curious student, you want to join as much sessions as possible. Design a greedy algorithm that finds the optimal list of sessions with the maximum number of sessions. Remember that you can be at only one session at the same time and you cannot leave any session before it is over. Implement your algorithm and analyze its worst-case running time. Explain your algorithm in your report file.

3-) You are given a set of integers S = [-1, 6, 4, 2, 3, -7, -5]. Design a dynamic programming algorithm to check whether there is a subset with the total sum of elements equal to zero. If the algorithm finds such a subset, then print the elements of that subset and terminate the algorithm. Implement your algorithm and analyze its worst-case running time. Explain your algorithm in your report file.

4-) Design a dynamic programming algorithm to find an alignment between two strings with minimum cost. The cost of the alignment is calculated using the following equation:

$$Cost = N * match\_score \ + M * mismatch\_score \ + K * gap\_score$$

The match, mismatch and gap scores in the equation are the weights for the matching, mismatching and indent operations, respectively. N, M and K are the count of matches, mismatches and gaps in the alignment, respectively. For example;

$$Sequence\ A\ = \ ALIGNMENT, Sequence\ B\ = \ SLIME$$

$$match\_score\ = 2, \qquad mistmatch\_score\ = -2, \qquad gap\_score = -1$$

$$Alignment\ of\ sequence\ A\ and\ B: \begin{cases} A\ L\ I\ G\ N\ M\ E\ N\ T \\ |\ \ |\ \ |\ \ |\ \ |\ \ |\ \ | \\ S\ L\ I - - M\ E \end{cases}$$

$$Cost = \big(4 * (+2)\big) + (1 * -2) + (2 * -1) = 4$$

Implement your algorithm and analyze its worst-case running time. Explain your algorithm in your report file.

5-) Suppose that you have an ancient computer system that needs to do $A + B$ operation to add numbers $A\ and\ B$. For example, the computer does 13 operation to add 5 and 8. You are given an array of integers to calculate the sum of its elements. Design a greedy algorithm to calculate the sum of the array with the minimum number of operations. Implement your algorithm and analyze its worst-case running time. Explain your algorithm in your report file.

Note:

* All implementations must be done using Python programming language (Python 3.6). All parts must be in one .py file.

* Try not to use external python libraries unless you have to.

* Write a driver function that includes at least one test for all your implementations.

* Prepare a report for your code explanations, complexity analysis etc. in pdf format. You can attach the scanned handwritten work you have, if you have any, and put them in the pdf file.

* Zip your report and .py file into a .zip file before submitting it.

* Submit your assignment via Moodle.

* Late submissions will not be accepted, so do not wait until the last minute.

*** Do your homework on your own; group studies will be considered as cheating.**