CSC 282/482 - Homework I (30 points)

Instructions: Please write your algorithms in clear pseudocode (or code). For problem 1 " Espresso Chain“ and for problem 2 "Lake Ontario Raid" you have to implement the solution in a programming language of your choice. In addition to the program, you also have to submit a README with instructions about execution, and at least 4 test cases you have used to test your program with the respective output you get and execution times. For all problems, you should argue the correctness of the algorithm, and analyze the order of running time. Please propose algorithms that are as efficient as possible. You will not get more than 2 points per problem for brute-force solutions.

1 Espresso Chain (4 + 2 points)

The powers at UR dining are thinking of launching a new chain of socially-distanced coffee shops along the Genesee River starting from Ford Street. All n possible locations are in the same line, with distances in increasing order $d_1, d_2, \ldots, d_n$ from Ford Street. The strategic plan is as follows:

- In each location there will be at most one coffee shop and the respective profit for location i is p_i, with $p_i > 0$ and $i = 1, 2, \ldots, n$.
- Any two coffee shops should be at least k miles apart with k a positive integer.

1. Describe an efficient algorithm for computing the maximum profit subject to the given constraints. Use the following subproblem: for all $1 \leq j \leq n$, $P(j)$ = maximum profit obtainable using only the first j locations.

2. Once you have solved the problem by *dynamic programming*, consider this 'fat oracle' approach:

```
Repeat:
    Pick the location i with highest profit p_i
    Remove all locations within k miles of location i
```

Do you think this will work? Explain your answer shortly.

2 Lake Ontario Raid (6 points)

The notorious pirate Blackbeard is planning to sail around Lake Ontario from Toronto to Rochester. There are n places where his crew can get some food, respectively in distances $m_1, m_2, \ldots, m_n$ from Toronto ($m_1 = 0$) with m_n (Rochester) as final destination. The pirates want to make as few stops as possible, but they can only make it to cover M miles after each stop before getting hungry again. Help the pirates[1] devise an algorithm to find the minimum number of stops they need to make in order to reach Rochester.

3 Ustá Grill

A traditional shish-kebab shop in downtown Rochester is having a long line of n dedicated customers patiently waiting to be served. The big chef Ustá is very experienced and he knows it takes c_i seconds to serve each customer i. The total waiting time is given by

$$T = \sum_{i=1}^{n} (waiting\ time\ for\ customer\ i)$$

Chef Ustá would like to minimize the waiting time for his customers. Can you help him by suggesting and describing an efficient algorithm for computing the optimal order in which to serve the customers?

4 Scripta Manent

Professor Archeoscriptus is working on some old scripts. In particular he is interested in identifying words that are the same whether read from right to left or from left to right, e.g. "NIΨONANOMHMATAMHMONANOΨIN". Your task is to devise an algorithm that helps the professor identify the *longest* of all such words that are a subsequence of a given sequence of letters $x[1..n]$.

Hint: You may want to look at $T(i, j)$ = length of longest word that is subsequence of $x[i..j]$.

5 Cash Manager

You are offered a position as cash manager at a luxurious casino in Niagara Falls. Your first task is to check if it is possible to give a change or not given an unlimited supply of coins with denominations $D = \{d_1, d_2, \ldots, d_n\}$ for an amount A. Armed with your experience from 282, you realize immediately this may be not possible for some denominations. The new boss asks for your help in devising a DP algorithm $O(nA)$ that decides if the change is possible or not. Help the boss solve the problem by writing an algorithm for the solution.

Hint: For $0 \leq s \leq A$, you can define $T(s) = true$ if it is possible to make change for s using coins of the given denominations.

[1] Illegal, but you know ...

6 k-Cash Manager

Note: This problem is only for graduate students.

A variant of the "Cash Manager" problem above with an additional constraint: you can use at most k coins.

Hint: You may need to use additional parameter(s).