COMP 3804 — Winter 2026 — Problem Set 5

Some useful facts:

1. $1 + 2 + 3 + \cdots + n = n(n+1)/2$.
2. for any real number $x > 0$, $x = 2^{\log x}$.
3. For any real number $x \neq 1$ and any integer $k \geq 1$,
$$1 + x + x^2 + \cdots + x^{k-1} = \frac{x^k - 1}{x - 1}.$$
4. For any real number $0 < \alpha < 1$,
$$\sum_{i=0}^{\infty} \alpha^i = \frac{1}{1-\alpha}.$$

Master Theorem:

1. Let $a \geq 1$, $b > 1$, $d \geq 0$, and
$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ a \cdot T(n/b) + \Theta(n^d) & \text{if } n \geq 2. \end{cases}$$
2. If $d > \log_b a$, then $T(n) = \Theta(n^d)$.
3. If $d = \log_b a$, then $T(n) = \Theta(n^d \log n)$.
4. If $d < \log_b a$, then $T(n) = \Theta(n^{\log_b a})$.

Question 1: In class, we have seen a data structure for the UNION-FIND problem that stores each set in a linked list, with the header of the list storing the name and size of the set, and each node storing a back pointer to the header. If we start with n sets, each having size one, then we have seen in class that, using this data structure, any sequence of $n-1$ UNION-operations can be processed in $O(n \log n)$ time.

Give an example of a sequence of $n-1$ UNION-operations, for which the algorithm takes $\Omega(n \log n)$ time.

Question 2: You are given two strings $X = x_1 x_2 \dots x_m$ and $Y = y_1 y_2 \dots y_n$ over some finite alphabet. We consider the problem of converting X to Y, using the following operations:

1. Substitution: replace one symbol by another one.
2. Insertion: insert one symbol.
3. Deletion: delete one symbol.

For example, if X = "logarithm" and Y = "algorithm", we can convert X to Y in the following way:

1. Start with "logarithm".
2. Inserting "a" at the front gives "alogarithm".
3. Deleting "o" gives "algarithm".
4. Replacing the second "a" by "o" gives "algorithm".

The *similarity* between the strings X and Y is defined to be the minimum number of operations needed to convert X to Y. For example, the similarity between X = "logarithm" and Y = "algorithm" is three, because X can be converted to Y using three operations, but not using two operations. If the string X has length m and the string Y is empty, then the similarity between X and Y is equal to m.

Give a dynamic programming algorithm (in pseudocode) that computes, in $O(mn)$ time, the similarity between the strings X and Y. Argue why your algorithm is correct. Follow the three dynamic programming steps that we have seen in class.

Question 3: Let $S = (a_1, a_2, \dots, a_n)$ be a sequence of n positive numbers. A subsequence T of S is called *awesome*, if for every i with $1 \leq i \leq n-1$, a_i and a_{i+1} are not both in T. In other words, whenever a number is in T, none of its neighbors in S is in T. The weight of the subsequence T is the sum of all numbers in T.

Give a dynamic programming algorithm that computes, in $O(n)$ time, the maximum weight of any awesome subsequence of T.

As always, justify your answer. Follow the three dynamic programming steps that we have seen in class.

Question 4: Zoltan is not only your friendly TA, he is also the CEO of *Zoltan Enterprises*. This company buys long copper wires, cuts them into subwires, and then sells these subwires. Zoltan Enterprises only buys long copper wires having integer lengths, and cuts them such that each subwire has an integer length.

Let n be a large integer and let $p_1, p_2, \ldots, p_n$ be a sequence of positive numbers. For each i with $1 \leq i \leq n$, Zoltan Enterprises sells a subwire of length i for p_i dollars.

Consider a copper wire of length n. Give a dynamic programming algorithm that computes, in $O(n^2)$ time, the maximum revenue that can be obtained by cutting the length-n wire into subwires.

As always, justify your answer. Follow the three dynamic programming steps that we have seen in class.

For example, let $n = 4$. Here are the different options to cut a length-4 wire:

- The wire is not cut. Then the revenue is p_4.
- The wire is cut into one subwire of length 1 and one subwire of length 3. Then the revenue is $p_1 + p_3$.
- The wire is cut into two subwires of length 1 and one subwire of length 2. Then the revenue is $2 \cdot p_1 + p_2$.
- The wire is cut into two subwires of length 2. Then the revenue is $2 \cdot p_2$.
- The wire is cut into four subwires of length 1. Then the revenue is $4 \cdot p_1$.