# CISC-471
# FALL 2019

## HOMEWORK 1

Please work on these problems and be prepared to share your solutions with classmates in class on Thursday, September 12. Assignments will **not** be collected for grading.

## Programming

Write a program in the language of your choosing (I recommend Python) and verify that it works on the sample data. For each problem be prepared to tell us why you think your algorithm is correct (whether you program worked on the sample data or not). Also provide an estimate of the time and space complexity of your algorithm.

**Frequent Words Problem:** A k-mer is defined as a string of length k. We define Count(Text, Pattern) as the number of times that a k-mer Pattern appears as a substring of Text. For example,

$$\text{Count(ACAACTATGCATACTATCGGGAACTATCCT,ACTAT)} = 3.$$

We note that Count(CGATATATCCATAG, ATA) is equal to 3 (not 2) since we should account for overlapping occurrences of Pattern in Text. We say that Pattern is a most frequent k-mer in Text if it maximizes Count(Text, Pattern) among all k-mers.

For example, “ACTAT” is a most frequent 5-mer in “ACAACTATGCATCACTATCGGGAACTATCCT”, and “ATA” is a most frequent 3-mer of “CGATATATCCATAG”.

**Frequent Words Problem** Find the most frequent k-mers in a string.
**Input:** A string Text and an integer k.
**Output:** All most frequent k-mers in Text.

**Sample Input:**
ACGTTGCATGTCGCATGATGCATGAGAGCT
4
**Sample Output:**
CATG GCAT

**Frequent Words with Mismatches:** Find the most frequent k-mers with at most d mismatches in a DNA string.

A k-mer is defined as a string of length k. We define Countd(Text, Pattern, d) as the number of times that a k-mer Pattern appears as a substring of Text with at most d mismatches. For example:

Countd(ACAACTATGCATACTATCGGGAACTATCCT,CTATG, 1) = 3,

as shown below.

ACAA**CTATG**CATA**CTATC**GGGAA**CTATC**CT

We say that Pattern is a most frequent k-mer in Text if it maximizes Countd(Text, Pattern,d) among all kmers.

**Input:** A DNA string Text as well as integers k and d.

**Output:** All k-mers, Pattern, maximizing the sum Countd(Text, Pattern,d)

**Sample Input:**

ACTATGCATACTATCGGGAACT
5 1

**Sample Output:**

CTATG CTATC ACTAT

## Problems

These questions come from *An Introduction to Bioinformatics Algorithms* by Neil C. Jones and Pavel A. Pevzner.

**Problem 2.2:** Write one (or two if you wish) algorithms that iterate over every index from (0, 0, . . . , 0) to $(n_1, n_2, \ldots, n_d)$. The output should be all values from (0, 0, . . . , 0) to $(n_1, n_2, \ldots, n_d)$. Suppose all of the values $n_1, n_2, \ldots, n_d = N$. If we say an output of (0, 0, . . . , 0) is one unit of output, how many units of output in terms of $N$ and $d$ are there? Now dropping the assumption that $n_1, n_2, \ldots, n_d = N$, how many units of output in terms of $n_1, n_2, \ldots, n_d$ and $d$ are there?

**Problem 2.3:** Is $\log n \in O(n)$? Is $\log n \in \Omega(n)$? Is $\log n \in \Theta(n)$?

**Problem 2.10:** Prove that $\sum_{i=1}^{n} i = n(n+1)/2$, for all $n \in \mathbb{N}, n \geq 1$.

**Problem 2.17:** There are $n$ bacteria and 1 virus in a Petri dish. Within the first minute, the virus kills one bacterium and produces another copy of itself, and all of the remaining bacteria reproduce, making 2 viruses and 2($n$ - 1) bacteria. In the second minute, each of the viruses kills a bacterium and produces a new copy of itself (resulting in 4 viruses and 2(2($n$ - 1) - 2) = 4$n$ - 8 bacteria; again, the remaining bacteria reproduce. This process continues every minute. Will the viruses eventually kill all the bacteria? If so, design an algorithm that computes how many steps it will take. How does the running time of your algorithm depend on $n$?

**Problem 2.18:** A very large bioinformatics department at a prominent university has a mix of 100 professors: some are honest and hard-working, while others are deceitful and do not like students. The honest professors always tell the truth, but the deceitful ones sometimes tell the truth and sometimes lie. You can ask any professors the following question about any other professor: ?Professor Y , is Professor X honest?? Professor Y will answer with either ?yes? or ?no.? Design an algorithm that, with no more than 198 questions, would allow you to figure out which of the 100 professors are honest (thus identifying possible research advisors). It is known that there are more honest than dishonest professors. It is also known that the professors know each other well and can tell with certainty whether a professor is honest or not.