

Assignment 1

Due Date: January 31, 2025
(10 points)

Instructions: Complete the assignment independently, as it is an individual task. Any instances of plagiarism will be met with strict consequences. Ensure to submit coding assignments in **Python (Jupyter Notebook)** by the due date. Each challenge should be a separate cell in the notebook for readability. Carefully follow all the provided guidelines.

Code Challenge 1: Implement pattern count function. Find the number of times a Pattern occurs in Text string. (1point)

Input: Strings *Text* and *Pattern*.

Output: *Count(Text, Pattern)*.

```
# Create your pattern_count function, along with any subroutines you need
def pattern_count(text: str, pattern: str) -> int:
    pass
```

Sample Input:

ACGTACGTACGT
CG

Sample Output:

3

Code Challenge 2: Solve the Frequent Words Problem. Given a string *Text* and an interger *k*, find all most frequent *k*-mers (*k*-sized patterns) in *Text*. You may create any supporting funtions/subroutines for your code. Note that there can be more than one most frequent patterns so your routine must return a list of patterns. (2 points)

Input: A string *Text* and an integer *k*.

Output: All most frequent *k*-mers in *Text*.

```
# Create your frequent_words function here, along with any subroutines you need
def frequent_words(Text: str, k: int) -> list[str]:
    pass
```

Sample Input:

CGTTTTGAACATTTTCAACAAGTTTTGCAACATTTT
4

Sample Output:

TTTT

Code Challenge 3: Solve the Minimum Skew Problem.

(2 points)

Minimum Skew Problem: Find all positions in a genome where the skew diagram attains a minimum.

Input: A DNA string *Genome*.

Output: All integer(s) i minimizing $Skew_i(Genome)$ among all values of i (from 0 to $|Genome|$).

```
# Create your MinimumSkew function, along with any subroutines you need
def minimum_skew(genome: str) -> list[int]:
    pass
```

Sample Input:

CCGGCCGG

Sample Output:

2 6

Code Challenge 4: Solve the Frequent Words with Mismatches Problem. Find the most frequent k-mers with up to d mismatches in a text. (5 points)

Input: A string *Text* as well as integers k and d . (You may assume $k \leq 12$ and $d \leq 3$.)

Output: All most frequent k -mers with up to d mismatches in *Text*.

```
# Create your frequent_words_with_mismatches function, along with any subroutines you need
def frequent_words_with_mismatches(text: str, k: int, d: int) -> list[str]:
    pass
```

Sample Input:

ACGTTGCATGTCGCATGATGCATGAGAGCT
4 1

Sample Output:

ATGT GATG ATGC

Final Challenge: Find a *DnaA* box in *Salmonella enterica*. **Note:** This challenge is optional and not currently graded.

Download File: [Salmonella_enterica.txt](#)