# BT-3051 Data Structures and Algorithms for Biology
# Assignment - 2

**Submission :** Since this is a coding based assignment, students need to submit their codes in a zipped folder. Your zip file should be named something like BTyyBxxx.zip, based on your roll number. This zip file must contain a single neatly typeset PDF of your solutions (named BTyyBxxx.pdf) including the codes used for each of the problems in a separate folder code with proper annotations.
**Deadline : Wednesday 28 August 2024**

**Instructions:**

a. Each question carries 6 marks. Total marks for this assignment - 30
b. Students are required to do just 5 of the 6 questions. You can pick any 5.

**Q1.** Huntington's disease is a neurodegenerative disorder caused by a mutation in the HTT gene on chromosome 4, where an increase in CAG repeats (more than 40, compared to the normal range of less than 35) leads to the production of a mutated Huntingtin protein. This protein causes gradual damage to brain cells.

Mohesh, a Bioinformatician at Dholakpur Pvt. Ltd., has been given a FASTA file containing the gene sequences of a client who wants to be tested for early signs of Huntington's disease. Knowing that abnormal CAG repeats can lead to abnormal GC content (the normal GC content for humans is between 30-60%), Mohesh decides to calculate the GC content of the HTT gene to determine if there is an abnormality.

Write a program to calculate the GC content of each gene in the FASTA file and use binary search to identify any outlier with abnormal GC content.

Note: a FASTA file is a text file for genome sequences representing nucleotides or amino acids.

**Q2. Nucleotide Sequence Matching.**
In a Genetic research lab, people are working with a large list of unique nucleotide sequences.
Each sequence is basically a combination of 'A','T','G','C' characters. They have come up with an interesting way to assign values to these nucleotides in the sequence. The nucleotide 'A' in the sequence possess the value **(i)+2**, whereas the nucleotide 'T' is valued **(2*i)+3**. The nucleotide 'G' in the sequence is equal to **(3*i)+4** and the nucleotide 'C' is valued **(4*i)+5**.
Here, i is the 1 based index of the nucleotide in sequence.

The sum of values of all the nucleotides of the given sequence is the spark **value** of that sequence. You are given the list of nucleotides and a value **k.** Write a program to find out if the sequence, whose spark value is equal to k, exists in the given list of sequences.
Return the index of the sequence if exists else -1.
**Note:** The solutions with time complexity of O(n^2) will not be accepted and it is guaranteed that the list of sequences given are in increasing order of spark value**.**

**Q3. (On OOPs)**

Lead compounds refer to potential drug compounds (mostly proteins) based on desired properties like the active site. The active site refers to a protein's enzymatic site of action to which other proteins might have affinity to bind to. This active site is exploited to create drugs that can bind to a drug target in the body inactivating the mechanism of disease.

Create a program using Classes and Objects where you define a Class that acts like a template for protein whose parameters are the protein name and its amino acid (AA) sequence. Define a function within the class that will count the number of times an active site occurs in the AA sequence given input of a desired active site. The function should also record the location of occurrence of the active site on the AA sequence. Comment on the efficiency of the data structure you have opted to use in the function.

You can use AA sequences of proteins from NCBI to test your code.

**Based on Tower of Hanoi**

**Q4. Multi-Disk Genetic Segregation with Crossover and Deterministic Mutation**

You are tasked with simulating the segregation of genetic material (DNA segments) across multiple chromosomes during cell division. The simulation involves two key rules:

**Crossover:** DNA segments on adjacent chromosomes can only be transferred if their size difference is within a specified threshold.

**Deterministic Mutation:** After a fixed number of moves, the DNA segment being moved will undergo a deterministic mutation (e.g., increasing or decreasing in size).

**Your task is to design an algorithm that:**

1. Transfers all DNA segments from the source chromosome to the target chromosome following the crossover and mutation rules.
2. Determines the minimum number of moves required to complete the segregation.

Constraints:

a. You are given num_chromosomes (number of chromosomes) and num_segments (number of DNA segments on the starting chromosome).
b. The crossover threshold defines the maximum allowable size difference for transferring a segment between chromosomes.
c. Mutations occur deterministically after a fixed number of moves.

**Requirements:**

I. Implement a function genetic_segregation(num_chromosomes, num_segments, crossover_threshold) that simulates this process.
II. Return the minimum number of moves required to complete the task.
III. Write pseudocode in comments explaining the steps of your algorithm.

Terms used :

**Chromosome:** A thread-like structure in the nucleus of a cell that carries genetic information. It is composed of DNA and proteins.

**NA (Deoxyribonucleic acid):** A molecule that carries genetic information. It is composed of nucleotides, which are made up of a sugar, a phosphate group, and a nitrogenous base.

**Gene:** A segment of DNA that codes for a protein or RNA molecule.Genetic material: The material that carries genetic information, which is typically DNA or RNA.Segregation: The separation of homologous chromosomes during meiosis, resulting in the distribution of genetic material to daughter cells.
**Crossover:** The exchange of genetic material between homologous chromosomes during meiosis.
**Mutation**: A change in the DNA sequence.

## Q5. Analyze Codon Usage

A codon is a sequence of three DNA bases that corresponds to a specific amino acid or stop signal during protein synthesis. Write a function to count the frequency of each codon in a given DNA sequence. This helps in understanding the gene's potential to code for proteins.

```
def codon_usage(dna_sequence):
"""
Calculates the frequency of each codon (set of three bases) in a DNA sequence.
:param dna_sequence: str, the DNA sequence to analyze
:return: dict, a dictionary with codons as keys and their frequencies as values
"""
pass
```

Example call:
codon_usage("ATGGCAATCAAGTCATTGGA")

Example output:
{'ATG': 1, 'GCA': 1, 'ATC': 2, 'AAG': 1, 'TCA': 1, 'TTG': 1, 'GAA': 1}

## Q6. Find Longest Homopolymer

A homopolymer is a sequence of identical bases in DNA (e.g., "AAAA" or "CCCC"). Write a function to find the longest homopolymer in a given DNA sequence. The function should return the base of the homopolymer and its length.

```
def longest_homopolymer(dna_sequence):
"""
Finds the longest homopolymer in a DNA sequence.
:param dna_sequence: str, the DNA sequence to analyze
:return: tuple, (base of the homopolymer, length of the homopolymer)
"""
pass
```

Example call:
longest_homopolymer("ATGGCAAAATCAAGGGGGG")

Example output:
('G', 7)  {Longest homopolymer is "GGGGGGG"}