

## Problem Set 4 Exercise #18: DNA String Computations

**Reference:** Lecture 11 notes

**Learning objectives:** Characters and Strings; Writing complicated program

**Estimated completion time:** 70 minutes

### Problem statement:

[CS1010 AY2010/11 Semester 1 Practical Exam 2 Exercise 1]

Consider the problem of comparing DNA sequences. A DNA sequence consists of a string where every character is one of the following 4 letters: 'A', 'C', 'G', or 'T'. For example, a DNA sequence might be the string "TCCTATTCTT".

#### Task A.

First, we would like to know what the **Hamming distance** is between several DNA sequences. The Hamming distance between two strings of equal length is the number of positions at which the corresponding characters are different.

For example, the Hamming distance between "ATCGT" and "CTAGG" is 3 because the two strings have different characters at 3 positions, namely positions 0, 2, and 4.

If there are more than 2 input strings, the Hamming distance needs to be computed between any pair of two strings. For example, given 3 input strings, the Hamming distance is computed between the 1<sup>st</sup> and the 2<sup>nd</sup>, the 1<sup>st</sup> and the 3<sup>rd</sup> as well as the 2<sup>nd</sup> and the 3<sup>rd</sup> strings.

Your task is to write a static method:

```
int[] hammingDistances(String[] dna)
```

which takes as input a list of DNA strings and returns an array of integers **dist** containing the Hamming distances among all pairs of strings.

The **dist** array should contain the Hamming distances in the following order: string pair 1<sup>st</sup> and 2<sup>nd</sup>, string pair 1<sup>st</sup> and 3<sup>rd</sup>, string pair 1<sup>st</sup> and 4<sup>th</sup>, ..., string pair 1<sup>st</sup> and N<sup>th</sup>, string pair 2<sup>nd</sup> and 3<sup>rd</sup>, string pair 2<sup>nd</sup> and 4<sup>th</sup>, ..., string pair 2<sup>nd</sup> and N<sup>th</sup>, etc., until string pair N-1<sup>th</sup> and N<sup>th</sup>.

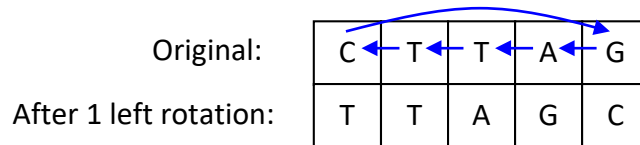
You should also fill in the **main()** method. The input and output of the program should be as shown in the sample runs section on the next page.

You may assume that all DNA strings are given in capitalized form and have equal lengths.

#### Task B.

Second, we would like to know whether the given DNA sequences are rotations of each other. A DNA sequence (let's call it DNA<sub>2</sub>) is a rotation of another sequence (say DNA<sub>1</sub>) if it can be produced by shifting the letters of the DNA<sub>1</sub> sequence either to the left or to the right while the characters wrap around at the beginning or the end of the string.

As an example, the following illustration shows a rotation where the second string “TTAGC” is a rotation of the original sequence “CTTAG” (the string was shifted to the left by one position and the first character was moved to the end).



Your task is to write a static method:

```
boolean[] isRotation(String[] dna)
```

which returns a boolean array **rotation** such that **rotation[i]** tells if **dna[i]** is a rotation of at least one other string sequence.

You may write additional methods as necessary.

Complete the skeleton program **PS4\_Ex18\_DNA.java** for the above two tasks.

#### Sample run #1:

```
Enter the number of DNA strings: 2
Enter string 0: ATTCC
Enter string 1: CCATT
The Hamming distance of pair 0 is: 5
String 0 is a rotation of another string
String 1 is a rotation of another string
```

#### Sample run #2:

```
Enter the number of DNA strings: 3
Enter string 0: AATCCCGT
Enter string 1: TCCAACGT
Enter string 2: CGTAATCC
The Hamming distance of pair 0 is: 5
The Hamming distance of pair 1 is: 7
The Hamming distance of pair 2 is: 6
String 0 is a rotation of another string
String 1 is not a rotation of any other string
String 2 is a rotation of another string
```

#### Sample run #3:

```
Enter the number of DNA strings: 4
Enter string 0: CCATTGCC
Enter string 1: ATGCCGCC
Enter string 2: ATTGCCCC
```

Enter string 3: **TTACCCGT**

The Hamming distance of pair 0 is: 5

The Hamming distance of pair 1 is: 6

The Hamming distance of pair 2 is: 7

The Hamming distance of pair 3 is: 3

The Hamming distance of pair 4 is: 5

The Hamming distance of pair 5 is: 5

String 0 is a rotation of another string

String 1 is not a rotation of any other string

String 2 is a rotation of another string

String 3 is not a rotation of any other string