Exercise 3: Searching

The purpose of this exercise is to give you more experience with using and modifying a search algorithm. Starter code for this exercise is provided in StringSearch.java. This exercise is due in the Exercise 3 dropbox by the due date specified on myCourses. To get full credit for this exercise, you must submit your code and a PDF showing screen shots with correct output for the two Test Cases shown below with each test case clearly marked.

Pattern Matching in DNA Sequence Data

Pattern matching is common in the field of bioinformatics. One of the fundamental pattern matching tasks is locating patterns within DNA sequences. DNA sequences consist of four possible characters, called bases; A, C, T, and G. Within a given DNA sequence, bioinformaticians typically search for all occurrences of a substring, called a target sequence. Today, using the code from the rabinKarp method in StringSearch.java, and the DNA sequence "GTTGCAGTTACTTATTATCTGAAAACCAGTTGATGTTAAGGAATACTCTGTC TAAGACAACATATGTAATAAAAATTATATATTCGTTGGGTTCTCTCGA", you'll be doing the following.

1. Rename the Rabin-Karp string search method to rabinKarpMultiple and modify the code so that it returns a data structure containing **all** of the locations of a given substring.

2. Write a countBases method that takes in a DNA sequence and outputs the total number of bases (A, C, T, G) in the sequence to the console. This method should not return anything and should call rabinKarpMultiple to assist in the calculations.

Test Cases

To prove functionality, you must show that the following test cases work properly. Your grade is out of a total of 10 points.

1. Output the locations of the substring "GTT" within the sequence. This output should be produced in the main method. (5 points)

2. Output the how many times each base occurs in the sequence. This output should be produced within the countBases method. (5 points)