COMP 6651: Algorithm Design Techniques
Fall 2017: Programming Assignment 2
Due: November 5, 2017 at midnight

1 Problem

Your task is to write and test two pattern matching algorithms. This kind of task is very important in many text processing applications, including in bioinformatics.

You are given as input two files. The first file is called `string.txt`, and is a string on the alphabet $\{A, \ldots, Z\}$. This string is called the *Text*. The text, could for example, be a genome. The length of the string is called m and is at most $400,000$. The second file is called `patterns.txt`. The first line of this file contains an integer called k. The next k lines each contain a string of length at most n where $1 \leq n \leq 200$. Each of the k strings is called a *Pattern*. We also know that $k \leq 10,000$.

Your job is to check, for each pattern, if it is a substring of the text, and if so, at what position. A pattern is a substring of the text at position i if $Pattern(j) = Text(i + j)$ for $0 \leq j \leq length(Pattern) - 1$.

Write your output into a file called `Output.txt`. For each pattern in the `patterns.txt` file, write the position where it is a substring of the text on a separate line. If the pattern is not a substring of the text, the corresponding entry should be -1. If there are multiple positions where the pattern occurs, you should output the first one.

To perform the pattern matching, you should implement two algorithms:

1. A straightforward algorithm that checks for each position i with $0 \leq i \leq m - n$ if the pattern occurs at position i.

2. An algorithm using suffix trees.

Some test cases will be provided on the course website. You should verify if your programs work on the test cases before submitting. Your code will also be tested on some larger input files which are not given to you in advance.

Run experiments on these two algorithms, and analyze and compare the running time of both algorithms as a function of the length of the text and the length of the pattern.

2 Requirements

You must submit a zip file containing the following three items:

- Source code for the two algorithms written in C#/C++/Java and
- A report with the results of your analysis.
- The data files you used in your analysis.

3 Programmer-on-duty

There will be a programmer-on-duty, Meghrig Terzian, available to help you with the assignment in the lab H-827 on Tuesdays and Thursdays from 5-8 p.m.