CMPT 451/813 — Assignment #2

2023–2024 T1

Due: *23:55, Friday November 3, 2023*

I will put a spot up on Canvas for you to upload the assignment. Please take all your files and zip them up into one file and submit the zip file. If something goes wrong for some reason and you can't submit on canvas, then just email me your solutions as attachments to mcquillan@cs.usask.ca. For the code questions, hand in code. For the other questions, you can either draw a solution digitally, or draw by hand, take a picture, and hand in a pdf.

The assignment is to be completed *individually*. Discussion with others is fine, but there cannot be passing of data or communicating lines of specific code (see details on syllabus).

Exercise 1

Let $\Sigma = \{a, b\}$. Construct the string matching automaton for the string $p = ababa$. The visual depiction of an automaton is sufficient and the formulaic description is not needed.

Exercise 2

Trace through Algorithm 2 of the notes with the string matching automaton from question 1, and the two different text patterns

1. $aababaa$,
2. $babaaa$.

Show at each step of the algorithm how the states are changing, and the output, if any.

Exercise 3

Construct a suffix trie for the word (text) $dbadba$.

Exercise 4

Let $\Sigma = \{a, b, c, d\}$. Construct a compact suffix tree for the string $t = dabdabc$. Use the strategy shown on slide 70 of the notes for Topic 4. Submit diagrams showing the suffix tree after step 1, step 2, and finally after step 3.

Exercise 5

We're going to use an implementation of suffix trees. This one worked well for me: https://pypi.org/project/suffix-trees/

Then download human chromosome 1: https://www.ncbi.nlm.nih.gov/nuccore/NC_000001.11?report=fasta. (If you're unfamiliar, hit 'send to', 'file'. This is a plain text file called a FASTA file, where the first line is a header, and the rest contains 60 nucleotides per line. FASTA files can be parsed with Biopython as follows: http://biopython.org/DIST/docs/tutorial/Tutorial.html#sec12 and can be easily converted to strings (needed for suffix trees).

Your code should put the first $10,000,000$ characters of chromosome 1 into a suffix tree. Then, it should iterate through all possible nucleotide sequences of length 9 (called 9mers), and print out a list of all 9mers that occur at least 1000 times (they can be overlapping) together with the number of times it occurs. The printed list should be ordered from most frequent to least frequent.