FIT2004: Lab questions for week 9

Objectives: This prac provides a platform for you to practise the formal concepts introduced during the lectures in weeks 7 & 8. Primarily, these concepts include pattern matching on strings and shortest-path algorithms on graphs.

1. The file **refText.txt** given in the supporting material directory (uploaded on Moodle in the week 9 section) contains a long **reference text** with 1 million characters plus the terminal '**$**' character at the end. The file **patterns.txt** contain $50,000$ short, variable-length **patterns**. (Note, the reference text and each of the $50,000$ patterns are strings over the alphabet A, C, G, T.)

   (a) Write a program that implements the Rabin-Karp's algorithm to identify how many times **each** of the $50,000$ patterns occur in the reference text. Note, Rabin-Karp employs a rolling hash function as given on the slide numbered as 40/51 in week 7 lecture slides. Using the notations on this slide, for this exercise, assume the characters '{A, C, G, T}' take the integer values $\{0, 1, 2, 3\}$ respectively, and assign the hash function parameters $z = 3$ and $q = 32,452,843$ (see slide 40 in `17.1_and_7.2_combined.pdf` on Moodle).

   (b) Also given in the supporting material directory is a file entitled **bwt.txt** that contains the Burrows-Wheeler Transform (BWT) of the reference text considered above. Using the same set of patterns (in **patterns.txt**), write another program that identifies how many times each pattern occurs using the backward search approach on BWT string.

   (c) Check if the answers from both these programs you have written match with each other. Having done this, did you observe any noticeable difference in the run times of the two programs you have written? If so, can this difference be explained?

2. Write a program implementing the Breadth-First Search (BFS) to find the single-source shortest paths in an **unweighted and undirected** graph. Your program should accept $|V|$ (the number of vertices) as a command line argument. Constrain this value to the range $1000 \leq |V| \leq 5000$. Generate a random (unweighted and undirected) graph with $|V|$ vertices – think carefully about how such a random graph must be generated. Assume the first vertex in this random graph is the source vertex. Run the BFS algorithm on this random graph instance and print out to a file (1) the path lengths to other vertices from the source, and (2) their corresponding paths.

```
-=o0o=-
  END
-=o0o=-
```