Data Structures and Algorithms – (COMP SCI 2C03)
Winter, 2021
Assignment-III

Due at 11:59pm on April 12th, 2021

- **No late assignment accepted.**
- Make sure to submit a version of your assignment ahead of time to avoid last minute uploading issues.
- Submit one assignment solution as a PDF file per group on Avenue.
- If the solution submitted by two groups is the same. Both teams will get a zero mark on the assignment.
- Present your algorithms in Java or Pseudocode (Pseudocode is preferred).
- It is advisable to start your assignment early.

This assignment consists of 8 questions, and is worth 40 marks.

1. How would you use Prim's and Kruskal's algorithms to compute the minimum spanning forest of an edge-weighted graph that is not connected? For this question only outline the procedure and discuss both algorithms as well as how your approach would differ depending on the algorithm used. No need for an implementation. Note: A minimum spanning forest is a collection of MSTs computed for each connected component.[5 marks]

2. Given an MST for an edge-weighted graph G, suppose that an edge in G that does not disconnect G is deleted. Describe how to find an MST of the new graph in time proportional to E. [4 marks]

3. Given a digraph with positive edge weights, and two distinguished subsets of vertices S and T, find a shortest path from any vertex in S to any vertex in T. Your algorithm should run in time proportional to $E \log V$, in the worst case. Only provide an outline/pseudocode for the algorithm [4 marks for the algorithm, 1 marks for explaining it runs in $E \log V$]

4. Given a weighted digraph, find a monotonic shortest path from s to every other vertex. A path is monotonic if the weight of every edge on the path is either strictly increasing or strictly decreasing. The path should be simple (no repeated vertices). Only provide an outline/pseudocode for the algorithm. No need to give an implementation. Hint : Relax edges in ascending order and find a best path; then relax edges in descending order and find a best path. [4 marks]

5. Develop a version of key-indexed counting that uses only a constant amount of extra space. Your developed version need not be stable. [4 marks]

6. Write an algorithm that, given two strings of equal length n, determines whether one is a cyclic rotation of the other, such as example and ampleex in time $O(n)$. A general explanation of why your algorithm runs in $O(n)$ time will suffice. You are not required to compute $T(n)$ and proof that $T(n) \in O(n)$. Note: A cyclic rotation of a string $s = uv$ is a string $s' = vu$. When $u = \varepsilon$, then $s = s'$. [5 marks for algorithm, 1 marks for explanation of it being in $O(n)$].

7. A tandem repeat of a base string u in a string s is a substring of s having at least two consecutive copies of u (non-overlapping). Develop a linear-time algorithm that, given two strings u and s, returns the index of the beginning of the longest tandem repeat of u in s. For example, your algorithm should return 3 when $u = abcab$ and $s = abcabcababcababcababcab$. [5 marks for algorithm, 1 marks for explaining that it is linear-time]

8. (i) Give the optimal Huffman code for the symbols a, b, c, d, e, f, g, h having frequencies $1, 1, 2, 3, 5, 8, 13, 21$ respectively. Observe that the frequencies listed are the first 8 Fibonacci numbers. [3 marks]

(ii) Generalize your answer for [i] to find the optimal Huffman code for n symbols, having their frequencies as the first n Fibonacci numbers, for a general n; that is, the i-th symbol has frequency F_i, where $1 \leq i \leq n$. [3 marks]