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

Due at 11:59pm on February 14th, 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 10 questions, and is worth 75 marks.

Question 1 Given a doubly linked-list (L), and its head and tail:

(a) Write an algorithm that outputs the reverse of the list L, as a new list, in $\Theta(n)$ time, where n is the length of the linked list. Your solution must update both the *next* and *prev* pointers appropriately. [5 marks]

(b) Explain your algorithm and its running time analysis. [5 marks]

Question 2 Given a non-negative integer array $A[1..n]$ (that is, A contains numbers from the set $\{0, 1, 2, 3, 4, \ldots\}$), the next smaller value of A (NSV_A) is an integer array of length n, where $NSV_A[i]$, $1 \leq i \leq n$, is defined as follows:

1. if $A[i] = 0$ then $NSV_A[i] = 0$
2. otherwise, $NSV_A[i] = j$, where
   (a) for every $h \in [1..j-1], A[i] \leq A[i+h]$, and
   (b) either $i + j = n + 1$ or $A[i] > A[i+j]$.

In short, $NSV[i]$ shows how far one should trace forward from A[i] to reach a smaller value than $A[i]$. Write an algorithm to compute the NSV array in $O(n)$ time, using the stack data structure. Below is an example of an NSV array. [6 marks]

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	0	1	3	3	7	7	1	5	5	0	2	2	6	6	0	4	4
NSV_A	0	8	4	3	2	1	3	2	1	0	4	3	2	1	0	2	1

Question 3 Implement the Queue data structure using the Stack data structure. For this question you need to implement all the Queue operations (Dequeue, Enqueue etc.) using ONLY the stack operations (PUSH, POP etc.). [6 marks]

Question 4 Using ONLY the definition of $O(f(n))/\Theta(f(n))$ prove that the following statements are TRUE. Your proofs using Limits will not get a mark:

(a) $(65n^4 + 2n + 3)/(n^4 + 1) = \Theta(n^4)$ [3 marks]

(b) $\log 2^{n^2} + n^2 + 1 = \Theta(n^2)$ [3 marks]

(c) $1 + 2 + 3 + \ldots + n = O(n^2)$ [3 marks]

Question 5 Using ONLY the definition of $O(f(n))/\Theta(f(n))$ prove that the following statements are FALSE. Your proofs using Limits will not get a mark:

(a) $(1000n^4 + n^2 + 4n)/(\log n) = \Theta(n^4)$ [3 marks]

(c) $\sqrt{n} = O(\log n)$ [3 marks]

Question 6 Given a sequence $< x_1, x_2, \ldots, x_n >$ of integers, we want to find a subsequence of consecutive items that give the largest sum. Let LI and

UI be the lower and upper indices defining such a subsequence of consecutive items, and let $MSum$ be the sum of items in this subsequence. Then your algorithm should report the following $< MSum, LI, UI >$. For example, if you are given the following sequence of integers $< -5, -3, 7, -2, 4, 6, -7 >$, then $< 15, 3, 6 >$ should be reported since the sum $x_3 + x_4 + x_5 + x_6 = 15$ gives the maximum. In the special case where all the integers are negative, $< 0, 0, 0 >$ should be reported.

(a) Give the simplest possible brute force $\Theta(n^3)$ algorithm for this problem. It should check all possible candidate sequences. [5 marks]

(b) Modify your algorithm from (a) to reduce the running time to $\Theta(n^2)$ [6 marks]

Question 7 Construct an algorithm that will sort four numbers in no more than 5 comparisons, regardless of the input. [5 marks]

Question 8 Give traces, in the style explained in class, showing how the numbers $9, 1, 5, 2, 11, 18, 7, 23, 13, 0, 4$ are sorted with top-down mergesort. [5 marks]

Question 9 Write an algorithm for the non-recursive version of quicksort based on a main loop where a subarray is popped from a stack to be partitioned, and the resulting subarrays are pushed onto the stack. Note : Push the larger of the subarrays onto the stack first, which guarantees that the stack will have at most $\log N$ entries. [7 marks]

Question 10 (a) Is an array that is sorted in decreasing order a max-oriented heap? Justify your answer. [5 marks]

(b) For $N = 16$, give arrays of items that make heapsort use the most number of compares, and the least number of compares. Explain your answer. [5 marks]