CS340 – Advanced Data Structures and Algorithm Design – Fall 2018
Assignment 6 – October 30, 2020

Dr. Sandra Zilles, Department of Computer Science, University of Regina

due November 6, 2020, 10.00pm

- Please submit both theory (a single pdf including program code, screenshots, etc.) and programming part (a single zip file) through UR Courses as usual.

Problem 1 (4 marks). Illustrate how Quicksort, using the median-of-three method for finding a pivot, works on input of the list 20, 13, 17, 25, 18, 2, 29, 14, 8.

Problem 2 (4 marks). Find a topological ordering for the graph described by the following adjacency matrix.

	A	B	C	D	E	F	G	H	I	s	t
A	∞	2	∞	∞	2	∞	∞	∞	∞	∞	∞
B	∞	∞	2	∞	∞	∞	∞	∞	∞	∞	∞
C	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	4
D	3	∞	∞	∞	3	∞	∞	∞	∞	∞	∞
E	∞	∞	2	∞	∞	3	∞	∞	3	∞	∞
F	∞	∞	1	∞	∞	∞	∞	∞	∞	∞	3
G	∞	∞	∞	2	1	∞	∞	6	∞	∞	∞
H	∞	∞	∞	∞	2	∞	∞	∞	6	∞	∞
I	∞	∞	∞	∞	∞	1	∞	∞	∞	∞	4
s	1	∞	∞	4	∞	∞	6	∞	∞	∞	∞
t	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Problem 3 (2+3 marks). Let $G = (V, E)$ be a graph with finitely many vertices. Prove the following statements.

1. If G has a cycle, then G has no topological ordering.
2. If G has no topological ordering, then G has a cycle.

Problem 4 (5+2 marks).

(a) Implement QuickSort, as explained in class, with varying pivot calculation methods and varying cutoffs. You may assume that all your input lists contain only integers, are duplicate-free, and are of length at most 100.

pivot calculation methods: pivot = first list element, pivot = second list element, pivot = element at middle position of list, pivot = median-of-three (as explained in class)

cutoffs: no cutoff, cutoff $c = 5$, cutoff $c = 10$, cutoff $c = 20$ (a cutoff here means an array size such that every array of size less than or equal to c will be sorted with Insertion Sort).

Add a counter to your implementation. Every time a comparison of two list elements is made, the counter is increased by one (also in the Insertion Sort routine).

(b) Test your implementation (all $4 \times 4 = 16$ variants) on three different lists of length 100 and, each time, report the total number of comparisons recorded in your counter. One of the input lists should be sorted in increasing order, one sorted in decreasing order, and one random. Show the 48 resulting values organized in a table, for better readability.