NANYANG TECHNOLOGICAL UNIVERSITY

School of Electrical and Electronic Engineering

Examination Marker's Report

Academic Year	2017/18	Semester	2

Subject Code/Title: EE2008/IM1001 Data Structures and Algorithms

Median Grade of Exam Paper *: EE2008 (D) / IM1001 (C)

** for the written final exam only; CA and other components are not included.*

Comments by Question

Q1. This question is comprised of 4 parts, namely (a), (b), (c) and (d).

(a) This question requires students to write an algorithm to find the first occurrence of the smallest element in a given array. Most of the students were able to answer this question correctly. However there were some who were not able to figure out the logic to write the algorithm correctly.

(b) This question requires student to simplify the sums of two given functions and to express the sum using the Big-Oh notations. Most of the students were able to answer this question correctly. However there were some who made minor mistakes in their calculations.

(c) This question test on the understanding of asymptotic notations. Students who have a good grasp of the definitions of the asymptotic notations are able to answer this question correctly.

(d) This question requires students to write a recursive algorithm to reverse the elements of a given array. This question proves to be challenging for some students. A common mistake made in the question is that students attempt to write a non-recursive algorithm to solve the problem.

Q2

(a) The overall performance for this part is average, as most students were able to understand and provide the pseudo code to determine the location of the rank r and do a proper shift down for the other elements. However, a small proportion of students shifted all entire array or some students did not take care & shifted it beyond the boundary. A small number did not understand the implementation of the vector is using an array and removing the element at rank r does not imply assigning zero to it.

(b) For this part (b) it was not a difficult question, it is generally much better and attempted by most of the student. They were able to write the proper pseudo-code for the implementation. A few students were not rigorous. They either forgot to set properly certain pointer fields like *node.next* and *node.prev*, or some of the implemented codes did not check to the end of the double-link list. This could be done with a while statement.

(c) However, for this part was generally not well done by the students. Some of the students did not use the recursive method. In addition, quite a number of students did not sum up the data field of the single-child node but implemented a simple counter instead. In addition, care was not taken to consider if the child/node was empty or not, nor the special cases when it had to explore both parts of the binary sub-trees. Candidate would have scored better marks by providing clear and concise pseudo codes.

Q3.

(a) Most students can't show the steps of Select algorithm clearly. Some students only show the results of partition algorithms instead of Select algorithm given in this exam paper.

(b) Most students can't show the counting sort step by step based on the given array. Some students can show up to the steps on how to get counting array.

(c) Very few students managed to write the algorithm correctly. Algorithms written by most students are not right for general cases.

Q4.

(a) This is a straight-forward application of Kruskal's algorithm to find a minimum spanning tree. Students who understood how Kruskal's algorithm works were able to get full marks for this part. Some students confuse this with Dijkstra's algorithm, which is to find a shortest path tree for a given source node.

(b) The answer is no. To fully justify the answer, you need to provide an example in which Kruskal and Prim's algorithms find different minimum spanning trees. A simple example with 3 vertices is sufficient. Students who attempted to use the graph in Figure 2 as an example came to the wrong conclusion as it turns out that in this particular graph, the minimum spanning trees found by both algorithms are the same.

(c) The question basically reduces to finding the shortest path tree from the given source vertex v and counting the number of vertices within k hops of v. Since the graph given is an unweighted graph, so there is no need to use Dijkstra's algorithm and using the breadth first search (BFS) procedure suffices. Note that BFS has better worst case time complexity than Dijkstra's algorithm. Some students attempted to use the depth first search (DFS) procedure, which is incorrect as DFS is not guaranteed to find the shortest path from v to another vertex.