ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION **WINTER SEMESTER, 2017-2018**

DURATION: 3 Hours **FULL MARKS: 150**

CSE 4303: Data Structures

Programmable calculators are not allowed. Do not write anything on the question paper.

There are **8 (eight)** questions. Answer any **6 (six)** of them.

Figures in the right margin indicate marks.

---

1. a) Define Complexity. Translate Q into its equivalent postfix expression P using stack. 9

$$Q = (A + B * C / D - E + F / G / (H + I))$$

b) On each iteration of its outer loop, insertion sort finds the correct place to insert the next item, relative to the ones that are already in sorted order. It does this by searching back through those items, one at a time. 4+4

i. Would insertion sort be speeded up by using binary search to find the correct place to insert the next item? Justify your answer.

ii. What is the running time for insertion sort when the array is already sorted in ascending order and descending order?

c) Discuss different types of memory allocation techniques with appropriate example. 8

2. a) Assume you have a stack with operations: *push(), pop(), top(), isEmpty()*. How would you use these stack operations to simulate a queue, in particular, the operations: *enqueue()* and *dequeue()*? 6

b) Consider the directed, weighted graph given in Figure 1. Even though the graph has negative weight edges, step through Dijkstra's algorithm to calculate shortest paths from A to every other vertex. Show your steps in a single table. Cross out old values and write in new ones, from left to right within each cell, as the algorithm proceeds. Also list the vertices in the order which you marked them visited. 13

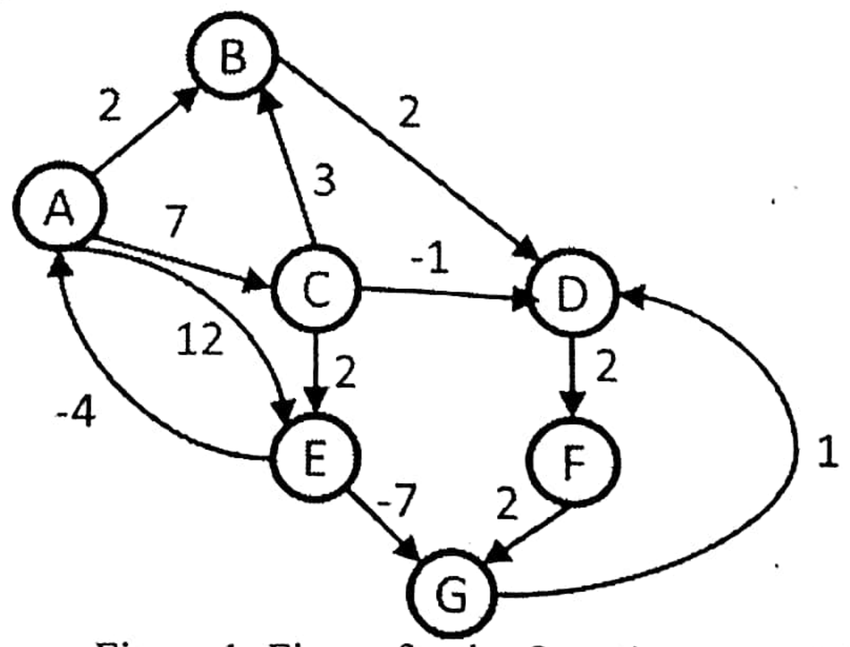

Figure 1: Figure for the Question no. 2 (b)

i. Here, Dijkstra's algorithm found the wrong path to some of the vertices. For just the vertices where the wrong path was computed, indicate both the path that was computed and the correct path.

ii. What single edge could be removed from the graph such that Dijkstra's algorithm would happen to compute correct answers for all vertices in the remaining graph?

iii. Determine all strongly connected components for the graph given in Figure 1.

c) What is garbage collection? Discuss how and when garbage collection takes place. 6

3. a) In the following questions, consider the list of numbers:

62 , 31 , 70 , 91 , 25 , 11 , 9 , 61 , 73 , 6

i. Show the resulting tree after inserting the numbers in the same order specified above into an initially empty minimum heap.

ii. Show the resulting tree after inserting the numbers in the same order specified above into an initially empty binary search tree.

iii. Use the binary search tree you created in (ii). What are the two possible binary search trees after 62 is deleted.

b) Draw an AVL Tree containing keys A, B, C, D, E, F, G such that a pre-order traversal visits the nodes in following order "E, C, B, A, D, F, G". Next, insert 'H' and 'I' (Balance the AVL tree if necessary). 7

c) Bob writes down a number between 1 and 1,000. Mary must identify that number by asking "yes/no" questions to Bob. Mary knows that Bob always tells the truth. If Mary uses an optimal strategy, then she will determine the answer at the end of exactly how many questions in the worst case. 6

4. a) Define Load factor in hashing. Write down some properties of a good hash function. Which type of Collision Resolution method is more appropriate when the load factor of the hash table is high? 8

b) What is the minimum and maximum number of nodes in a complete and perfect binary tree of height h? 5

c) What is hashing? Consider a hash table of size 11 storing entries with integer keys. Suppose the hash function is $\boldsymbol{h\,(k)\ =\ k\ mod\ 11}$. Insert, in the given order, entries with keys **0, 1, 6, 7, 10, 22, 21** into the hash table to resolve collisions using: 12

i. Separate Chaining

ii. Linear-Probing with $h_i(k) \Rightarrow (h(k) + i)\ mod\ 11$, where $f(i) = i$

iii. Quadratic-Probing with $h_i(k) \Rightarrow (h(k) + i)\ mod\ 11$, where $f(i) = i^2$

iv. Double-Hashing with h as the hash function and h_2 as the second hash function $h_i(k) \Rightarrow (h(k) + ih_2(k))\ mod\ 11$, where $h_2(k) = 5 - (k\ mod\ 5)$

5. a) How many strongly connected components does a directed acyclic graph (DAG) of n vertices have? Possible answers are 0, 1, n, or "it depends". Point out one of these answers and fully justify it. 7

b) The frequencies of characters used in an arbitrary message are as follows: 10

A : 7, B : 9, C : 11, D : 14, E : 18, F : 21, G : 27, H : 29, I : 35, J : 40

Show the complete Huffman tree for these characters and represent the values in binary for node A, C, and J of the Huffman tree using Huffman coding. (In Huffman tree, every left branch is coded with 0 and every right branch is coded with 1). 8

c) What is the best asymptotic ("big-O") characterization of the following functions:

i. $f(n) = 2^5 + 5n^3 \log(n) + 2^6 n^2 + 100\, n^4$

ii. $f(n) = 1000n^2 + 16n + 2^n$

iii. $f(n) = n + (n-1) + (n-2) + \cdots + 3 + 2 + 1$

iv. $f(n) = 2^{10} + 3^5$

6. a) Let T be a full binary tree with 2011 nodes. How many internal nodes does T have? Suppose you want to find the shortest distance from s to some particular vertex (rather than to all vertices reachable from s). What would you do? 4+4

b) Analyze the complexity of bucket sort and radix sort. Briefly mention the limitations of these two sorting algorithms. 4+4

c) Perform topological sort on the graph given in Figure 2 and write down the sorted list of nodes. If there is more than one valid topological sort order, write down all of them. 9

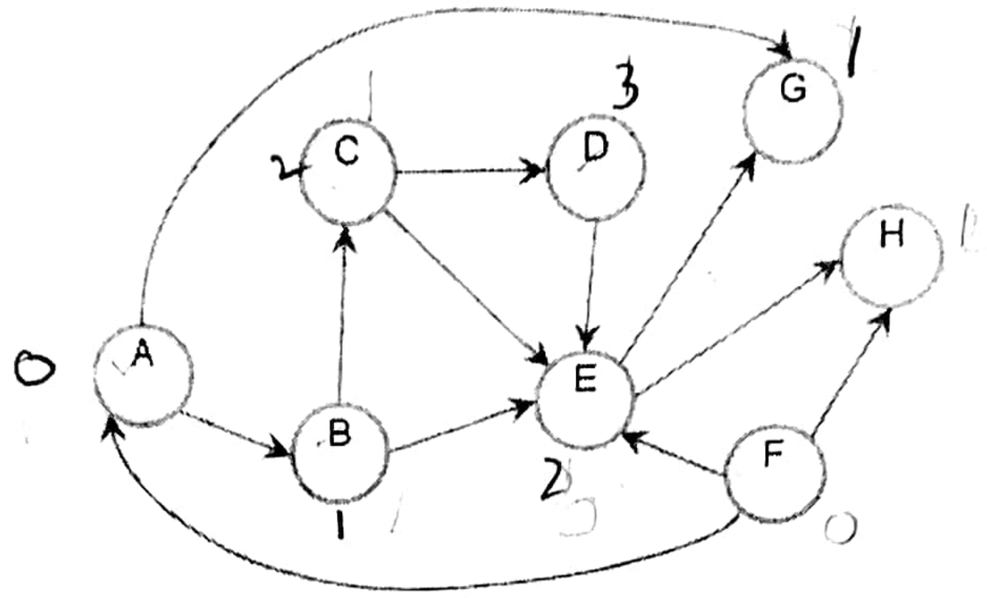

Figure 2: Figure for the Question no. 6 (c)

Next, for Figure 2, list the nodes in the order they would be visited in a depth-first search and breadth-first search of the graph starting at A. When choosing a node to explore next, break ties in favor of the alphabetically least.

7. a) Analyze the following scenarios regarding Prim's and Kruskal's algorithm and justify your answer with "yes/no". You are allowed to do some pre-processing work (like modifying the input values if necessary) before applying the algorithms. But you cannot modify the basic algorithm. 12

   i. Both algorithms will always return the same Minimum Spanning tree (MST).
   ii. Both algorithms can produce Minimum Spanning Tree with an undirected graph where the weights can be either positive or negative.
   iii. You can compute a maximum spanning tree, namely the spanning tree that maximizes the sum of edge costs.

   b) Analyze the following code of Figure 3 and find out the complexity in terms of Big-O notation step by step. (try to make the upper bound tighter) 5

```
void complexity(){
    int n, val;
    for ( int j = 4; j < n; j = j+2 ){
        val = 0;
        for ( int i = 0; i < j; ++i){
            val = val + i * j;
            for ( int k = 0; k < n; ++k){
                val++;
            }
        }
    }
}
```

Figure 3: Figure for the Question no. 7 (b)

   c) Explain critical path for a directed graph with suitable example. How to check from preorder, inorder or postorder traversal sequence whether a given Binary Tree is BST or not? 4+4

8. a) Write down one basic application and asymptotic run time for the following algorithms, on a graph with n vertices and m edges: 8

   i. Bellman-Ford algorithm
   ii. Breadth-first search (BFS)
   iii. Depth-First Search (DFS) with Adjacency matrix
   iv. Topological sort with Adjacency list

   b) Use Prim's algorithm starting at node A to compute the Minimum Spanning Tree (MST) of the graph given in Figure 4. In particular, write down the edges of the MST in the order in which Prim's algorithm adds them to the MST. Use the format (node 1; node 2) to denote an edge. Next, write down the edges in the order in which Kruskal's algorithm adds them to the MST. 7+4

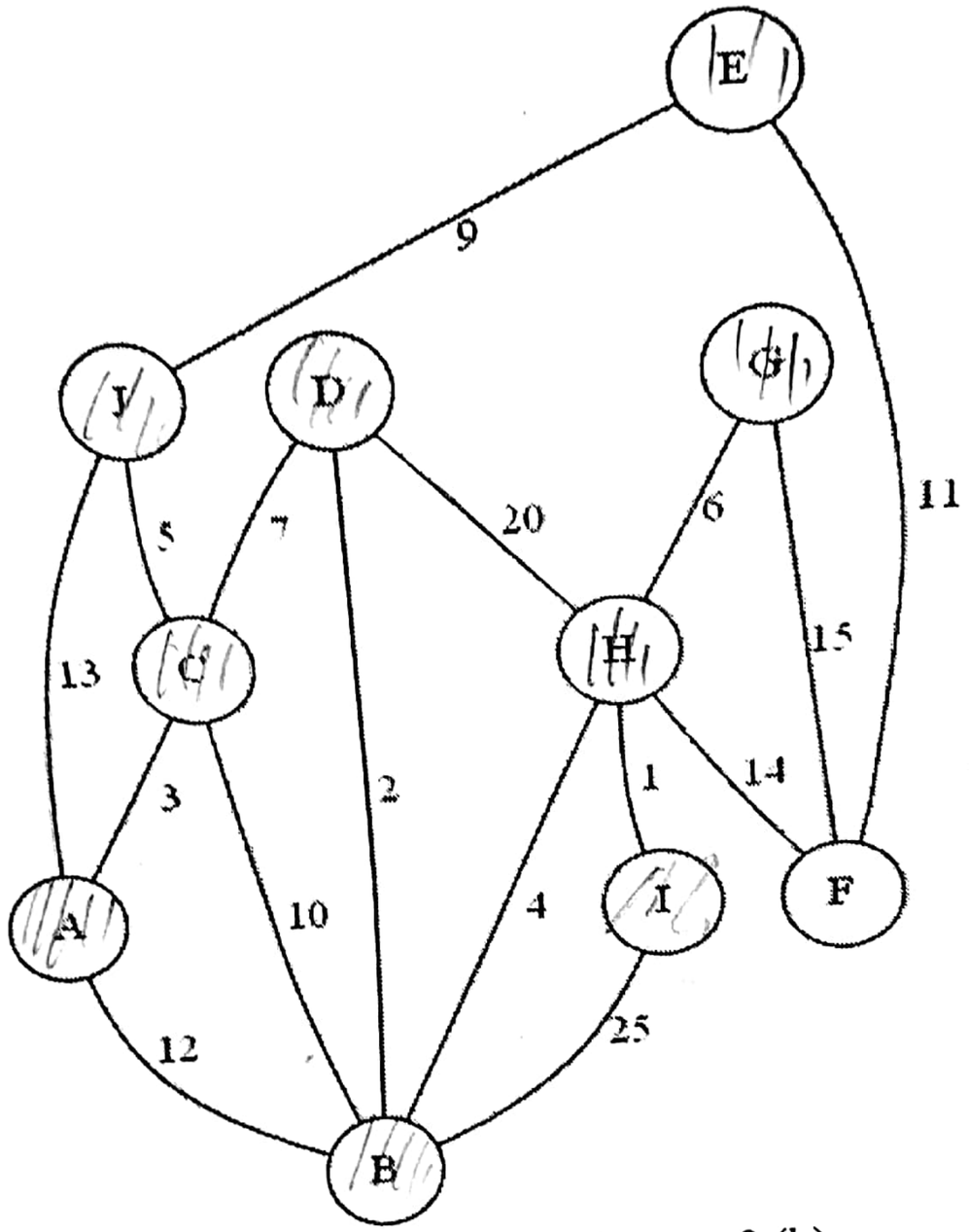

Figure 4: Figure for the Question no. 8 (b)

c) Consider a sorted circular doubly-linked list where the head element points to the smallest element in the list. Now write down the asymptotic complexity for the followings: 6

i. Finding the smallest element in the list
ii. Finding the largest element in the list
iii. Determining whether a given element *e* appears in the list
iv. Deleting a given element *e* in the list (not including the cost of finding it)