B.Sc. Engg. SWE 4th Semester

07 March 2024 (Afternoon)

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION | **SUMMER SEMESTER, 2022-2023**

DURATION: 1 HOUR 30 MINUTES | **FULL MARKS: 75**

CSE 4403: Algorithms

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

Answer **all 3 (three)** questions. Figures in the right margin indicate full marks of questions with corresponding COs and POs in parentheses.

---

1. a) Define algorithm. Write 4 real-life application areas of algorithms. — 1 + 4

   b) Apply 2D peak finding algorithm on the 8×8 matrix in Table 1. You do not need to rewrite the matrix in every step: — 10 (CO1) (PO1)

Table 1: Matrix for Question 1.b

3	2	1	5	5	2	7	5
7	4	4	8	7	7	0	5
2	2	6	4	8	9	7	7
9	8	7	2	2	7	0	7
5	1	3	7	2	6	4	1
2	5	5	0	2	6	2	8
7	6	9	8	7	8	6	2
5	9	8	8	8	9	1	3

   c) Analyze the run-time complexity of the build_heap() function with proper mathematical reasoning. — 5 (CO4) (PO2)

   d) Analyze the worst case run-time complexity of Insertion Sort with respect to the number of comparison(s). — 5 (CO4) (PO2)

2. Consider there is a number of cities in Bangladesh and they are inter-connected with many roads having different distances.
   Scenario 1: You want to travel from a particular city to some other city using the 3^{rd} best shortest path.
   Scenario 2: The government of the country wants to remove some of the roads keeping all the cities connected and minimize the total distance considering the remaining roads.

   a) Considering the scenarios, formulate the problem into a weighted graph of n nodes and m edges. — 3 (CO2) (PO2)

   b) Considering the scenario 1, modify the Dijkstra's Algorithm to find the 3^{rd} best shortest path. — 12 (CO3) (PO3)

   c) Considering the scenario 2, design an algorithm to find the roads that are needed to be removed. — 10 (CO3) (PO3)

3. a) From the recurrence shown in Equation 1 analyze the run-time complexity of ternary search algorithm. — 5 (CO4) (PO2)

$$T(N) = T(2N/3) + \Theta(f(n)) \quad (1)$$

b) Write the step-by-step pseudo-code of Counting sort and Bucket sort algorithms. 5 + 5 (CO1) (PO1)

c) Modify the topological sort pseudo-code shown in Algorithm 1 to find the critical path and critical time. 10 (CO3) (PO3)

```
   Input: G(V,E)                                          // Directed Graph
   Output: T                                  // Topolically sorted item list
1  Function TopSort(G(V,E)) :
2      T ← empty list                                  // T stores the topsort
3      Z ← empty queue          // Z stores vertices with in-degree 0
4      in ← dictionary mapping all vertices to 0        /* in stores current
        in-degree of each vertex */
5      for each v ∈V do
6          for u adjacent to v do
7              increment in[u]                                  // initialize in
8      for each v ∈V do
9          if in[v] = 0 then
10             add v to Z                                       // initialize Z
11     while Z is not empty do
12         v ←Z.front()
13         Z.removeFront()
14         append v to T                  // get next vertex for the topsort
15         for u adjacent to v do
16             decrement in[u]                                  // update in
17             if in[v] = 0 then
18                 add v to Z                                   // update Z
19     return T
```

Algorithm 1: Pseudo-code of Topological Sort for Question 3.c