United International **University**

Department of Computer Science and Engineering

Course: CSI 227 Algorithms | Trimester: Summer 2018

Final Exam | Marks: 100 | Time: 2 hours

There are FOUR questions. Answer ALL questions.

1. a) Let $G = (V, E)$ be a graph. Then write algorithms that **5 × 2**

   i) finds the maximum degree Δ of G when G is represented by *Adjacency Lists*,

   ii) finds the minimum degree δ of G when G is represented by *Adjacency Matrix*.

   Analyze the running time of the two algorithms. **3 × 2**

   b) Write an algorithm that topologically sorts a graph. Find a topological-sort for the graph shown in Figure 1. **3 + 6**

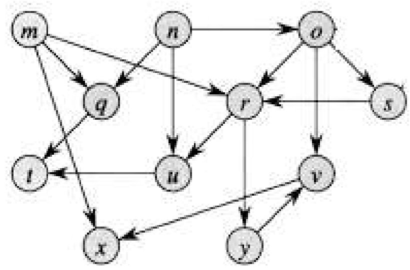

Figure 1

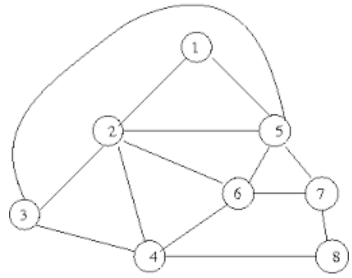

Figure 2

2. a) Dijkstra's algorithm to the single source shortest path problem is shown in Figure 3. What would be the time-complexity of the algorithm (show line by line analysis) for an input graph G if **10**

   i) *an array* is used instead of priority queue, and

   ii) *adjacency list* is used instead of adjacency matrix.

```
Dijkstra(G)
  for each v ∈ V
    d[v] = ∞;
  d[s] = 0; S = ∅; Q = V;
  while (Q ≠ ∅)
    u = ExtractMin(Q);
    S = S U {u};
    for each v ∈ u->Adj[]
      if (d[v] > d[u]+w(u,v))
        d[v] = d[u]+w(u,v);
```

Figure 3

```
BFS(G, s)
  initialize vertices;
  Q = {s};
  while (Q not empty)
    u = Dequeue(Q);
    for each v ∈ u->adj
      if (v->color == WHITE)
        v->color = GREY;  v->d = u->d + 1;  v->p = u;
        Enqueue(Q, v);
    u->color = BLACK;
```

Figure 4

   b) Suppose you are given weighted undirected graph G and its minimum spanning tree T. Give an algorithm that finds the second best minimum spanning tree. Analyze the running time of your algorithm. **10**

   c) Consider the graph shown in Figure 2. Assign weights to each of the edges such that the MST of this graph consists exactly of the edges (1, 5), (3, 4), (3, 5), (4, 6), (4, 8), (6, 7), and (2, 4). **5**

3. a) What do you understand by "collision" in the context of hashing? Collisions are resolved by the methods of 'Chaining' and 'Open Addressing'. Write the advantages and disadvantages of Chaining and Open Addressing methods. **1 + 6**

b) What would the resultant forests be after calling UNION(d, g) and then UNION(k, m) on the disjoint-sets forest of the following figure? You must use the *union-by-rank* and the *path-compression* heuristics. **4 × 2**

c) A simple version of the Breadth-First Search (BFS) algorithm is given in Figure 4. Modify the algorithm in a manner such that it will find the number of components in G. Assume that G is an undirected disconnected graph. What is the running time of your algorithm? **10**

4. Answer <u>*any one*</u> of the following *two* questions:

a) I) Briefly explain how probe sequences are generated by Linear Probing, Quadratic Probing, and Double Hashing techniques of open addressing. **4 × 3**

II) Consider an open-addressing hash table as shown below. The table already contains four data items, and other empty slots contain NIL. Assume that collisions are handled by Quadratic probing using the hash function $h(k, i) = (\boldsymbol{h'}(\boldsymbol{k}) + i^2) \bmod 13$, where $\boldsymbol{h'}(\boldsymbol{k}) = (k + 7) \bmod 13$. By showing detailed calculations, redraw the table after **13**

(i) *insert* 47; (ii) *insert* 64; (iii) *delete* 12 (replacement with NIL); (iv) *search* 38.

Although 38 had been present at the hash table, your search would fail here. Explain how one can modify the operations such that this will be prevented.

0	1	2	3	4	5	6	7	8	9	10	11	12
70						12	38					44

b) I) What do we mean when we say a problem is in *P, NP, NP-Complete*? **3**

II) Explain with a clear example and simulation of the Bellman-Ford algorithm that how may it be used to detect negative cycles in a directed graph. Use a directed graph with 4 vertices and 5 edges. **7**

III) Assume that the size of a data item and of a pointer is 50 and 4 units respectively. With $m = 200$ slots and $n = 50$ data items, what is the total amount of memory required, if data items are stored by i) direct-address table hashing; ii) hashing with chaining. **8**

IV) Suppose that using some unknown algorithm, you have obtained the following shortest path from a source s to a destination t: $s, d, f, g, h, j, k, f, r, t$. The graph contains all positive edge-weights. Even though you do not know what algorithm has been used, mention why is this shortest path incorrect. **7**