United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam Total Marks: **40** Fall 2021

Course Code: CSE 2217 Course Title: Data Structure and Algorithms II

Time: 2 hours (writing) + 15 minutes (Download/Upload)

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

There are **SEVEN** questions. **Answer all of them**. Show full simulation or solved graph figures wherever necessary. Figures in the right-hand margin indicate full marks.

First determine X and Y correctly for your student ID and write it down. Use these values in Questions 2, 3, 4 and 5.

For example, a student with ID: **011 142 001**

A B

A=142, **B**=1

```
X = 1 + (142 mod 6) = 1+4 = 5
Y = 1 + (1 mod 5) = 1+1 = 2
```

1. **(a)** Following is a course prerequisite dependency graph (Figure 1). Use topological sort to order the courses. Show details. **[3]**

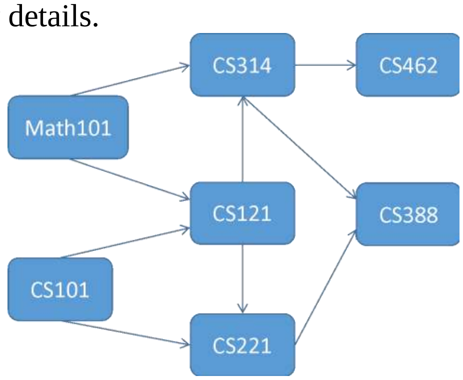

Figure 1: Course dependency graph

[1]

(b) Explain how we can detect if a graph has any cycle or not using DFS?

2. **(a)** Table 1 shows the parent array of a Disjoint set (Rooted tree implementation). Perform the following operations sequentially **using path compression and union-by-rank heuristic:**

i. Draw the disjoint set forest **[2]**

ii. What will be returned by Find-Set(Y), and Find-Set(6)? **[1]**

iii. Redraw the forest after Union(6, 9). **[1]**

iv. Redraw the forest after Union(9, Y). **[1]**

Index	0	1	2	3	4	5	6	7	8	9	10	11
Parent	0	0	1	0	3	4	7	8	8	9	X	Y

Table 1: Disjoint set (Rooted tree implementation)

(b) Write a pseudocode for the Disjoint set data structure that prints all the elements that are present in the set where an element E is present. You can assume that the disjoint set has N elements from 1 to N and Make-set, Find-set and Union operations are already implemented. **[2]**

3.

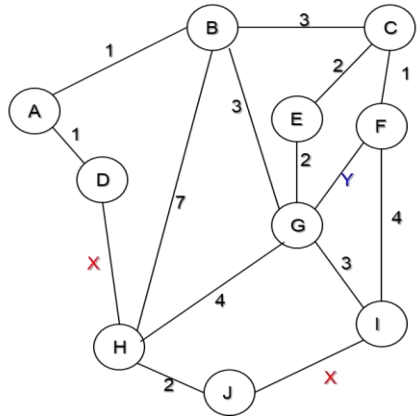

Figure 2: Question 3(a)

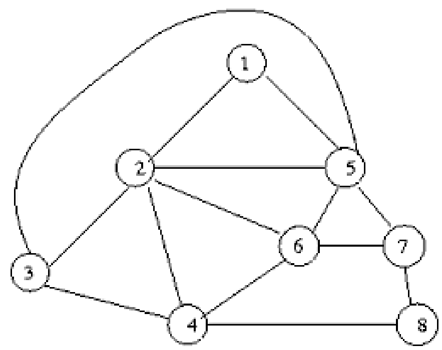

Figure 3: Question 3 (b)

(a) Find the Minimum Spanning Tree (MST) for the given graph in Figure 2 using Prim's algorithm. Show the details of your calculation. **[3]**

(b) Consider the graph shown in Figure 3. 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) **[3]**

4.

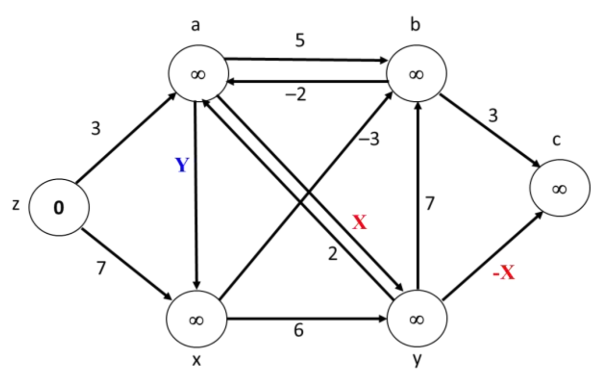

Figure 4: Directed Graph for Question 4(a)

(a) Find the Shortest path tree for the graph in Figure 4, where the source is vertex z. Show the details of your calculation. **[3]**

(b) Suppose 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 this shortest path is incorrect. **[1]**

(c) Explain why a priority queue instead of a regular queue (FIFO) is used in Dijkstra's algorithm. Give an example of a graph on which Dijkstra's algorithm is unable to find shortest paths if a regular queue is used. **[2]**

5. **(a)** What is Primary Clustering? Which Collision Resolution technique resolves this suffering? **[1]**

(b) Consider an open-addressing hash table as shown below (Table 2). The table already contains three data items. Assume that collisions are handled by the following hash function.

$$h(k, i) = \left(h'(k) + i\,h2(k)\right) mod\ 13,$$
$$where\ h'(k) = (2k + 7)\ mod\ 13\ and\ h2(k) = (k + 5)\ mod\ 13.$$

By showing calculations, redraw Table 2 and show following operations

i. Insert 70 **[1]**
ii. Insert 83 **[1]**
iii. Insert X **[2]**
iv. Search 99 **[1]**

0	1	2	3	4	5	6	7	8	9	10	11	12
				44	12				51			

Table 2: Open Addressing Table

6. Consider the mentioned notations
Text, $t = \text{“}LITTLEKITTENS\text{”}$
Pattern, $p = \text{“}IT\text{”}$
Modulo, $q = 13$
Hash of a string xy, $h(xy) = (x + y)\ mod\ q$

character	E	I	K	L	N	S	T
numerical value	5	9	11	12	14	19	20

Table 3: Numerical value for characters

Now answer the following questions using Rabin-Karp String matching algorithm

(a) Show the indices of valid hit and spurious hit (if any) using the aforementioned hashing function. **[2+1]**

(b) What is the time complexity of this algorithm? Explain how it improves the naïve string-matching algorithm? **[2]**

7. **(a)** What do we mean when we say a problem is in P, NP, NP-Complete? Show the relationship among these three classes of problems. **[2+2]**

(b) Suppose we know that problem A is NP-complete, and we want to show that another problem B is NP-complete. Do we reduce A to B, or reduce B to A? Justify your answer. **[2]**