United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam Total Marks: **50** Summer 2024

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

Time: 2 hours

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

There are **five** questions. **Answer all of them**. Show full simulation/tabulations wherever necessary. Figures in the right-hand margin indicate full marks.

1.

(a) Suppose a Mayor wants to connect 6 cities (A, B, C, D, E, F) by constructing roads among them. Constructing a road between two particular cities requires a certain cost. One can travel in both directions using a road. The cost to construct a road between two particular cities is given in the following table. Find the **minimum cost** required to connect all cities by constructing roads. Use an efficient algorithm and **clearly mention the name of the algorithm** you are using. **[5]**

Roads between cities	A and B	B and D	C and D	C and E	A and C	E and F	A and F	A and D
cost	7	12	10	8	10	12	15	10

Table 1 : Road construction Costs between cities

(b) If all the edge weights are positive in a Graph, then how can you find **Maximum** Spanning Tree using Kruskal or Prim's algorithm? Provide your idea by giving an example. **You do not need to write any pseudocode** [*A maximum spanning tree is a spanning tree of a graph with the highest possible sum of edge weights.*] **[2]**

(c) Which Minimum Spanning Tree (MST) algorithm will you use between Prim and Kruskal for the following types of graphs? **Justify** mentioning the time **complexity** of the algorithms. **[2]**

i) Dense Graph ii) Sparse Graph

2.

(a) Table 2 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(13), and Find-Set(3)? (Redraw the forest) **[1]**

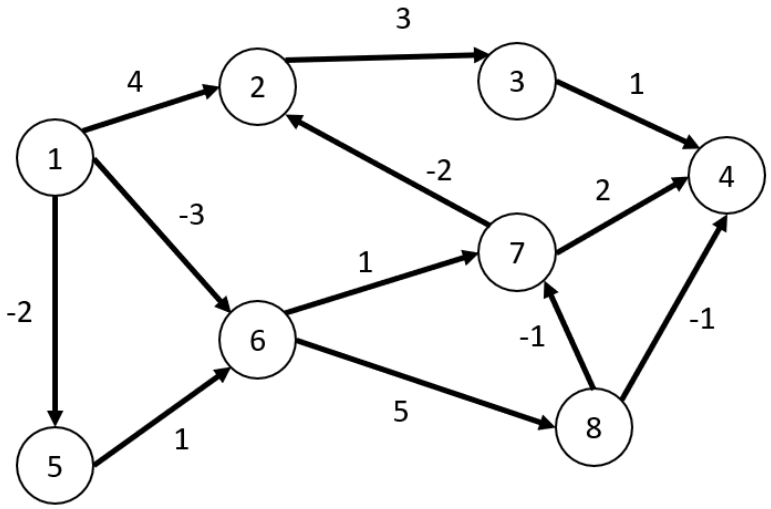
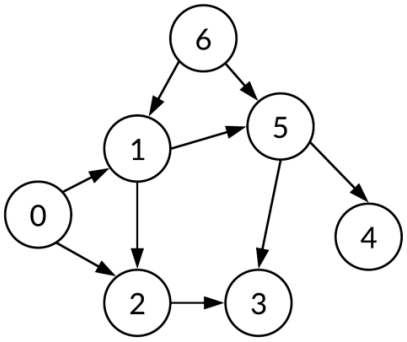
iii. Redraw the forest after Union(6, 10). Show necessary steps **[3]**

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Parent	1	1	1	3	4	5	7	7	7	9	2	7	12	13

Table 2 : Parent array of a Disjoint set

(b) Why do we use the heuristics: **union-by-rank** and **path-compression** in Disjoint-Set data structure? **Explain** your answer through **appropriate example(s)**. **[3]**

3 (a) Dijkstra's algorithm does not work correctly on graphs with negative edges, even if there are no negative cycles. Use a graph with negative edges (but no negative cycles) to show why **[3]**

	Dijkstra's algorithm **can fail**. Draw a **diagram to explain** your answer. **(b)** Determine the **shortest paths** from **vertex 1** to all other vertices in the graph given in Figure 1. Show all the necessary calculations step-by-step. Additionally, **print** both the **shortest distances** and **the corresponding paths for each vertex**.  Figure 1 : A weighted directed graph **(c) Justify** the following statement with **an appropriate example**: "The single source shortest path problem satisfies the optimal substructure property". **(d)** "If all the edges of a graph have equal weights, then **Breadth First Search** (**BFS)** gives the Minimum Spanning Tree and Single Source Shortest Path more efficiently than any other algorithms you have studied in this course" - do you agree or not? Show logical reasoning. **You don't have to show any simulation using BFS.** **(e)** Consider the following graph in Figure 2. Use **topological sort** starting from **node 2** to find a linear ordering of the nodes. Show details.  Figure 2 : Graph for topological sort	**[5]** **[2]** **[2]** **[3]**
4	**(a) Explain** a scenario in which the Rabin-Karp algorithm **performs no better than** the naive string matching algorithm.	**[2]**

	(b) You are tasked with finding all occurrences of a specific pattern within a text using a hash-based pattern-matching approach. The characters in the text and pattern are represented as follows: 	char	a	b	c	d	f
---	---	---	---	---	---
code	2	3	5	7	11	 The hash function is defined as follows: $$hash(s) = \left(\sum_{i=0}^{m-1} (s[i] * d^i) \, mod \, q \right) mod \, q$$ where, m = size of the string you want to hash d = m + p, where p = 3 q = 131 For string s = "bfd" , s[0] = 'b', s[1] = 'f', s[2] = 'd'. Given this information, find all occurrences of the pattern **"abd"** in the text **"abfabd"**. Show the **indices** of both **valid hits** and **spurious hits** (if any) using the aforementioned hash function.	**[5]**
5	**(a) Define** three types of probing in **open addressing with proper examples**. State pros and cons of each probing mechanism. **(b)** Consider an **open-addressing** hash table as shown below (Table 3). The table already contains four data items. Assume that collisions are handled by the following hash function. ***h(k,i) = (h' (k) + i h''(k)) mod m,*** where *h'(k) = (2k+7) mod m* and *h''(k) = 1 + (3k mod m); m = 13* By showing calculations, redraw the table and show following operations i. Insert 52 ii. Insert 70 iii. Delete 98, Replace with NIL iv. Search 40 | Index	0	1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	---	---	---	---	---
Value					83	12			98	27			| Table 3: Open Addressing Table **(c)** What is **primary clustering**? Do you think there is any 'primary clustering' in Table 3? Justify your answer.	**[3]** **[1]** **[2]** **[1]** **[1]** **[2]**