



United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam

Total Marks: 40 Summer 2022

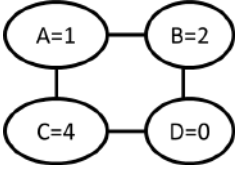
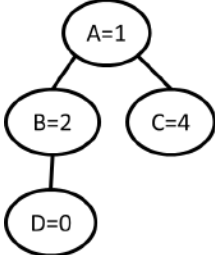
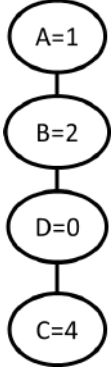
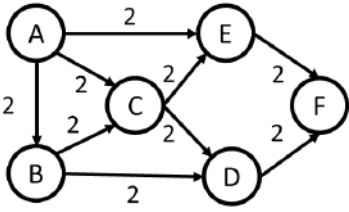
Course Code: CSE 2217

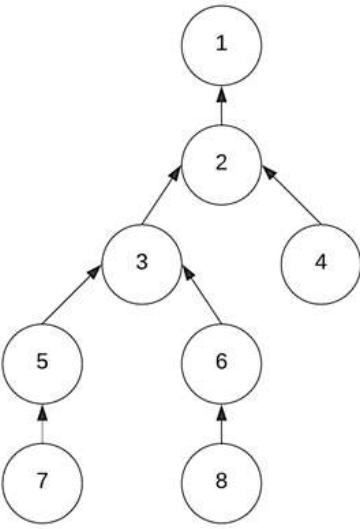
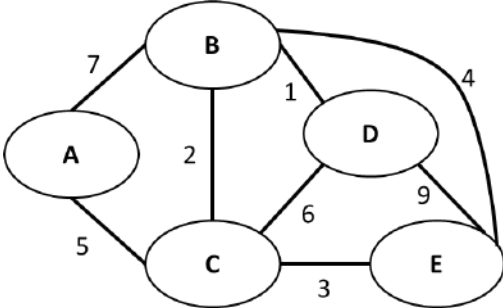
Course Title: Data Structure and Algorithms II

Time: 2 hours

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

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

1.	<p>(a) Fig. 1(a) is a graph where nodes A, B, C and D contain values 1, 2, 4 and 0 respectively. The BFS and DFS trees of the graph are in Fig. 1(b) and Fig. 1(c) respectively. So the paths of the BFS tree form the numbers 120 and 14 while the paths of the DFS tree form the number 1204.</p> <p>Given this formulation, what will be the graph if the BFS tree forms the numbers 1246, 1247 and 13 and the DFS tree forms the numbers 12467 and 13? You can assume that each node will have a unique value within 0 to 9 and the graph will contain no more than 10 nodes.</p> <div style="display: flex; justify-content: space-around; align-items: flex-end;"> <div style="text-align: center;">  <p>Fig. 1(a)</p> </div> <div style="text-align: center;">  <p>Fig. 1(b)</p> </div> <div style="text-align: center;">  <p>Fig. 1(c)</p> </div> </div>	[2]
	<p>(b) Can you find the shortest path from A to C in the following graph (Fig. 2) using BFS? Explain your answer briefly.</p> <div style="text-align: center;">  <p>Fig. 2. A directed graph</p> </div>	[1]
	<p>(c) “Breadth First Search” can create a minimum spanning tree for unweighted graphs. Explain why this statement is true.</p>	[2]

2	(a) What is union-by-rank in the Disjoint-Set Union data structure? Why do we need this heuristic? Justify your answer.	[2]
	(b) Write the pseudocode of Union Sets and Find Sets operation of Disjoint-Set Union data structure, <i>assuming the path-compression and union-by-rank heuristics</i> .	[3]
	(c) Show the status of Fig. 3 after each of the following operations: find_set(3) , find_set(6) , find_set(5) , find_set(3) . Note that the operations are executed sequentially, and <i>you must use the path-compression heuristic</i> .	[2]
 <pre> graph BT 1((1)) 2((2)) --> 1 3((3)) --> 2 4((4)) --> 2 5((5)) --> 3 6((6)) --> 3 7((7)) --> 5 8((8)) --> 6 </pre> <p>Fig. 3. Disjoint set rooted tree</p>		
3	(a) What will be the runtime of <i>Kruskal's algorithm</i> assuming edges are sorted using bubble sort? Note that bubble sort takes $O(n^2)$ time to sort n elements.	[1]
	(b) Apply <i>Prim's algorithm</i> to find the MST from the graph in Fig. 4 , assuming vertex D as the root. Show details and clearly draw the MST you have found.	[4]
 <pre> graph TD A --- 7 B A --- 5 C B --- 2 C B --- 1 D B --- 4 E C --- 6 D C --- 3 E D --- 9 E </pre> <p>Fig. 4. An undirected graph</p>		
	(c) “ <i>Kruskal's algorithm</i> grows a forest of trees and eventually forms the MST.” - Explain this statement with an example.	[2]
4	(a) A graph contains the vertices {A, B, C, D, E, F, G} and the shortest path from A to B is $A \rightarrow E \rightarrow C \rightarrow D \rightarrow F \rightarrow G \rightarrow B$. Is it possible to find the shortest path from E to F from given data? Justify your answer.	[2]

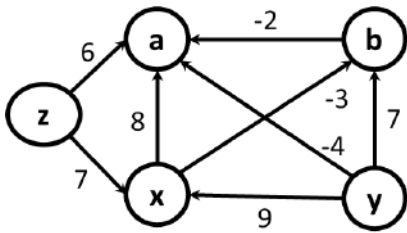
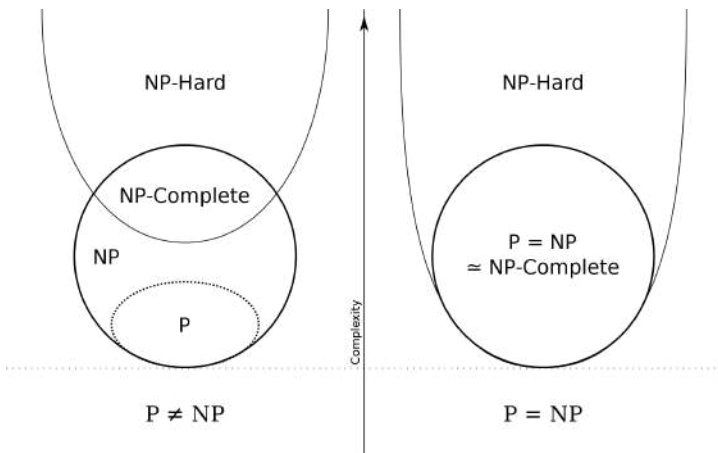
	<p>(b) Find the Shortest path tree for the given graph in Fig. 5, where the source is vertex z. Show the details of your calculation. Justify how the shortest path tree satisfies <i>optimal substructure property</i>. [3+2]</p> <div style="text-align: center;"></div> <p style="text-align: center;">Fig. 5: A directed graph $G(V,E)$</p>															
5	<p>(a) What are the benefits of a good Hash Function? [1]</p>															
	<p>(b) Which data structure is best suited for implementing Chaining in a HashTable: Binary Search Tree or Linked List? Explain with time complexity comparisons between the both for operations such as Insertion, Search and Deletion. [3]</p>															
6	<p>(a) Explain the working of the Rolling Hash Function in the Rabin-Karp algorithm using a suitable numerical example. [3]</p>															
	<p>(d) Given a test string: fcdabdea and a pattern: dac, use Rabin-Karp’s algorithm to find whether the pattern exists in the string or not. You must show the Hash values and calculations for all the substrings. The values of each character are given below. [2]</p> <table border="1" style="margin-left: auto; margin-right: auto;"><tr><td>char</td><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td><td>f</td></tr><tr><td>value</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>	char	a	b	c	d	e	f	value	1	2	3	4	5	6	
char	a	b	c	d	e	f										
value	1	2	3	4	5	6										
7.	<p>(a) Write the definitions of the following terms: [3]</p> <ul style="list-style-type: none">i. NP-Completeii. Piii. NP-hard															
	<p>(b) What does the following diagram (Fig. 6) represent? Explain. [2]</p> <div style="text-align: center;"></div>															

Fig. 6: Question 7(b)

Fig. 6: Question 7(b)

