Date of Examination: 08. 06. 2021

AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Department: Computer Science and Engineering
Program: B.Sc. in Computer Science and Engineering
Semester Final Examination: Spring 2020
Year: 2nd Semester: 2nd
Course Number: CSE2207
Course Name: Algorithms

Time: 3 (Three) Hours **Full Marks: 60**

Use single answer script

Instructions:	i)	Answer script should be hand written and should be written in A4 whitepaper. You must submit the hard copy of this answer script to the Department when the university reopens.
	ii)	You must write the following information at the top page of each answer script: **Department:** **Program:** **Course no:** **Course Title:** **Examination:** **Semester (Session):** **Student ID:** **Signature and Date:**
	iii)	Write down Student ID, Course number and put your signature on top of every single page of the answer script.
	iv)	Write down page number at the bottom of every page of the answer script.
	v)	Upload the scan copy of your answer script in PDF format through provided **google form** at the respective course site (i.e., **google classroom)** using institutional email within the allocated time. Uploading clear and readable scan copy (uncorrupted) is your responsibility and you must cover all the pages of your answer script. However, for clear and readable scan copy of the answer script student should use only one side of a page for answering the questions.
	vi)	You must avoid **plagiarism,** maintain **academic integrity and ethics**. You are not allowed to take any help from another individual and if taken so can result in stern disciplinary actions from the university authority.
	vii)	Marks allotted are indicated in the **right margin.**
	viii)	Assume any reasonable data if needed.
	ix)	Symbols and characters have their usual meaning.
	x)	Before uploading, rename the PDF file as **CourseNo_StudentID.pdf** e.g.,CSE2207_180204777.pdf

The answer script **(one single PDF file)** must be uploaded at designated location in the provided **Google Form link** available in the Google classroom.

There are 7 (Seven) Questions. Answer any 5 (Five).

Question 1. [Marks: 12]

	You have an array of records of *n* students. Each record contains Student_ID, Student_Name and Total_Marks. The array is sorted in ascending order based on Student_ID. Now you are assigned a task to sort the array in non-ascending order based on Total_Marks. If more than one student obtains equal marks then they retain their previous ordering. Mention the name of one sorting algorithm which must be used for this purpose. Justify your answer. Write down the algorithm and calculate its computational complexity.	**[12]**

Question 2. [Marks: 12]

a)	Write down the Kruskal's algorithm to find a Minimum Spanning Tree. A graph consists of following edges: *AB=3, AC=2, AE=@, BC=1, CD=#, CE=4, DE=6* and *BF=5*. Apply Kruskal's algorithm on the graph to find a Minimum Spanning Tree. ["@" = First Non zero digit and "#" = Second Non zero digit from the right side of your Student ID].	**[6]**
b)	Apply dynamic programming approach to solve the travelling salesperson problem starting from node "#" for the graph represented as the following cost adjacency matrix. ["#" = 2 for even Student IDs and "#" = 3 for odd Student IDs]. **Node** | **1** | **2** | **3** | **4** **1** | 0 | 10 | 15 | 20 **2** | 5 | 0 | 9 | 10 **3** | 6 | 13 | 0 | 12 **4** | 8 | 8 | 9 | 0	**[6]**

Question 3. [Marks: 12]

a)	Apply any programming strategy to find the Longest Common Subsequence (LCS) and its length for the sequences (*A, M, M, A, A, B, B, A*) and (*M, A, B, A, B, A*).	**[6]**
b)	Write an algorithm for solving 0-1 knapsack problem. Apply appropriate programming approach to solve the 0-1 knapsack problem for the following 4 items where knapsack size is 6. **Item** | **Weight** | **Value** 1 | 1 | 12 2 | 3 | 14 3 | 4 | 25 4 | 2 | 10	**[6]**

Question 4. [Marks: 12]

a)	What is an asymptotic tight bound? Explain with an example. Show that, there is no tight bound for ***n***!.	**[6]**
b)	Solve the following recurrences: i) $T(n) = c$ if $n = 1$, $T(n/10)+T(9n/10)+cn$ if $n > 1$; using recurrence tree method. ii) $T(n) = 2T(n/2) + n^2$; using master method.	**[6]**

Question 5. [Marks: 12]

a) Write an algorithm that constructs the optimal Huffman code for a given set of characters with their frequencies. Apply greedy approach to generate Huffman code for the following character sequence: **[6]**

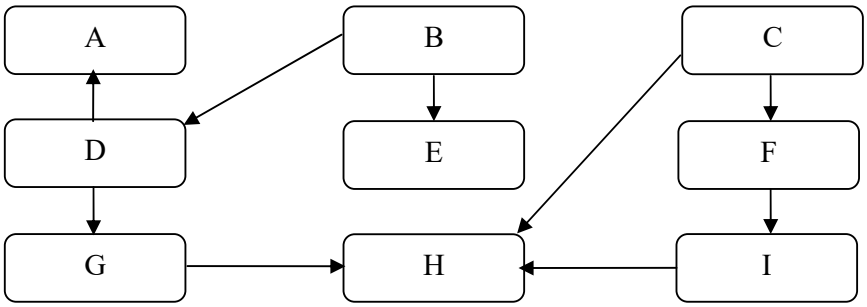
"AHSANULLAH"

Show the character frequency table and Huffman tree to generate the Huffman code for each character. For example, the frequency of the character 'H' is 2 in the above character sequence.

b) Draw the state space tree to show all the backtracking steps to find the first subset of {2, 4, 3, 5} that sum to 8. **[6]**

Question 6. [Marks: 12]

a) Write a topological sorting algorithm and then sort the following Directed Acyclic Graph (DAG) topologically. **[6]**



b) Using Floyd-Warshall algorithm find only the cost of shortest paths between each pair of nodes of the graph represented as the following cost matrix. ["#" = Second Non zero digit from the right side of your Student ID]. **[6]**

Node	**1**	**2**	**3**	**4**
1	0	3	5	∞
2	∞	0	∞	#
3	∞	6	0	∞
4	2	∞	- 3	0

Question 7. [Marks: 12]

a) "To solve the single source shortest paths problem, Dijkstra's algorithm may not produce correct result on a weighted, directed graph containing negative-weight edges" – justify the statement with an example. **[6]**

b) A directed graph consists of following edges: $st = 4$, $sy = 5$, $tx = 6$, $ty = @$, $yx = 2$, $yt = 3$, $xz = -4$, $xt = -1$, and $zy = 4$. Apply Bellman-Ford algorithm on the graph to find all the shortest paths from the source vertex "s". ["@" = First Non zero digit from the right side of your Student ID]. **[6]**