Roll Number:_____

Thapar Institute of Engineering & Technology

Department of Computer Science and Engineering

MID SEMESTER EXAMINATION

B. E. **(2nd Yr. COE/CSE)**

19th March, 2024

Tuesday, Time– 3:00 To 5:00 PM

Time: 2 Hours

Max Marks: 45 (Weightage: 30)

Course Code: UCS415

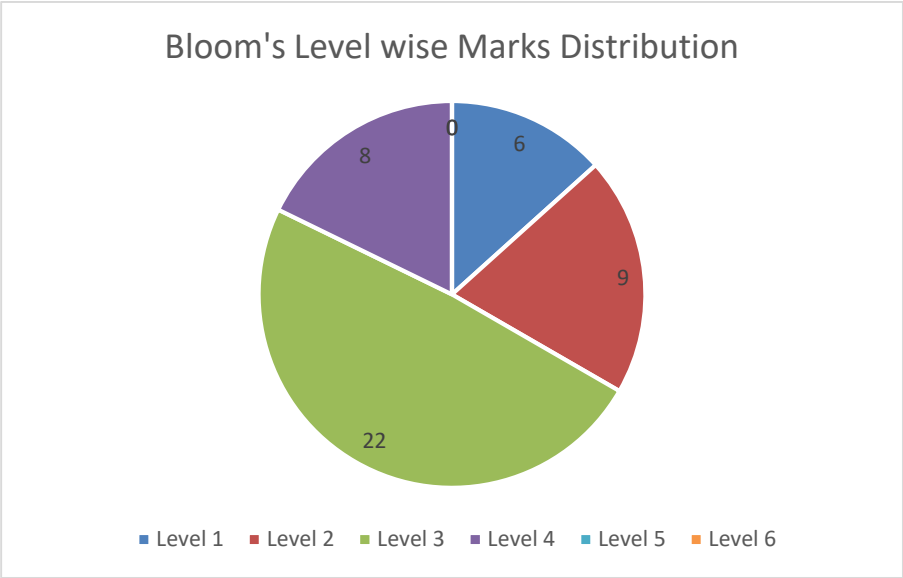
Course Name: Design and Analysis of Algorithms

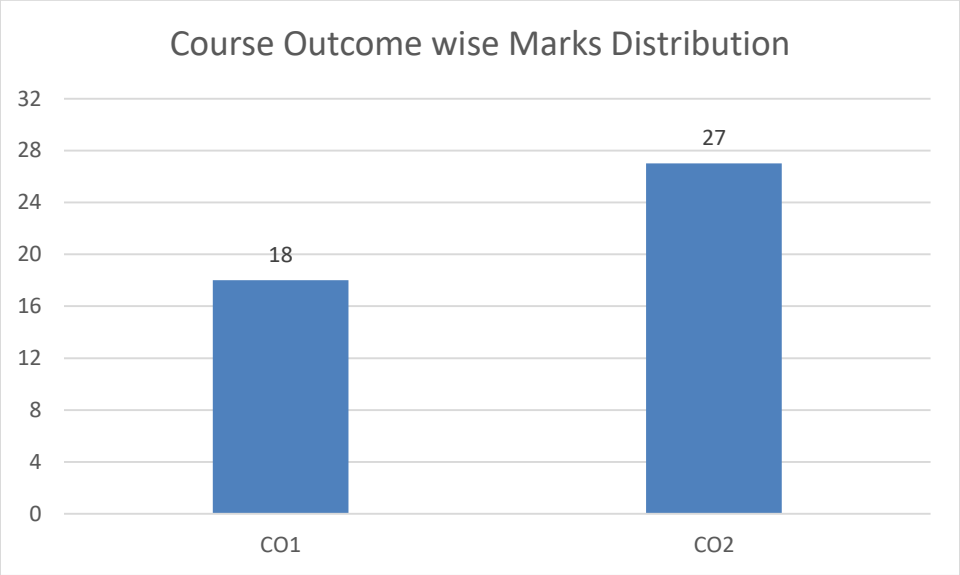
Name of Faculty: Tarunpreet Bhatia, Sumit Sharma, Manisha Singh, Swati Sharma, Mansi Sharma, Shubhra Dwivedi, Shruti Aggarwal

Note: Answer all sub-parts of each question at one place. Assume missing data (if any).

Q.No	Questions	Marks	CO	BL
Q.1	(a) You are a cashier at a store, and you need to give change to a customer for a specific amount using the fewest possible coins. You have an unlimited supply of coins with denominations of 1, 5, 10, 12, 25, and 50 cents. Suppose a customer needs change of 64 cents. What will be the greedy choice property selected by the cashier, and what coins will be given to the customer? Will the greedy algorithm always provide the optimal solution for every change amount using the given denomination of coins? Justify your answer with the help of a suitable explanation.	(4)	CO2	L2
	(b) Solve the following recurrence relation using the master method: $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}$, where T(1) = 1	(3)	CO1	L1
Q.2	(a) You are given a one dimensional array $A = \{-2, 3, 6, -5, 7\}$ that may contain both positive and negative integers. You need to determine the subarray of numbers which has the largest sum using divide and conquer approach. Show the intermediate steps.	(3)	CO1	L1
	(b) There are 2 sorted arrays $A[1..n]$ and $B[1..n]$ of size n each. Design an algorithm of $O(\log n)$ time complexity to find the median of the union A ∪ B (i.e. an array of length $2n$).	(4)	CO1	L4
Q.3	(a) Consider two strings S1 = "BCFFBF" and S2 = "CFCFFB". Compute the LCS table using bottom-up dynamic programming approach and mark the entries in the LCS table followed to determine LCS between S1 and S2 and print it. Also, write an algorithm to print LCS between S1 and S2 from the LCS table.	(4+2)	CO2	L3
	(b) Suppose we have n skiers with heights given in an array $P[1..n]$, and n skis with heights given in an array $S[1..n]$. Design an efficient greedy algorithm to assign a ski to each skier, so that the average difference between the height of a skier and his/her assigned ski is as small as possible. The algorithm should compute an assignment array $A[1..n]$, indicating that each skier i should be assigned to ski $A[i]$ such that the expression $\frac{1}{n}\sum_{i=1}^{n}\left\|P[i] - S\left[A[i]\right]\right\|$ is as small as possible. Also, apply your algorithm to the given input P = [3, 2, 5, 1] and S = [5, 7, 2, 9] to find the assignment array A.	(2+2)	CO1	L4

Q.4	A file contains the following characters with the frequencies as shown in **Table 1**. **Table 1** Character: C, D, E, M, O, U Frequency: 11, 2, 20, 8, 12, 3 If Huffman Coding is used for data compression (assume 0 is assigned to the left edge and 1 is assigned to the right edge), create a Huffman tree and answer the following questions: (a) Encode the text MOOD. (b) Decode the text 0001100011. (c) What will happen if Huffman code of character 'O' is reversed?	(5)	CO2	L2
Q.5	You are given 6 assignments of 6 different subjects and each assignment has a deadline and associated marks (as shown in **Table 2**). You need one day to complete one assignment and you can work on one subject daily. Marks are obtained if the assignment is completed before its deadline. Your task is to apply a greedy algorithm to find the optimal schedule for obtaining maximum marks. What will be the maximum marks you will get? **Table 2** Subjects: UCS401, UCS402, UCS403, UCS404, UCS405, UCS406 Deadline: 22.3.2024, 21.3.2024, 22.3.2024, 23.3.2024, 21.3.2024, 23.3.2024 Marks: 30, 20, 50, 10, 25, 15	(4)	CO1	L3
Q.6	Suppose we are designing a program to translate text from English to French. For each occurrence of each English word in the text, we need to look up its French equivalent. We could perform these lookup operations by building a binary search tree (BST) with n English words as keys and their French equivalents as satellite data. Moreover, some words in the text might have no French translation. Because we will search the tree for each individual word in the text, we want the total time spent searching to be as low as possible. We could ensure $\Theta(log_2 n)$ search time per occurrence by using balanced BSTs such as AVL, red-black trees etc. However, words appear with different frequencies in the text, and a frequently used word may appear far from the root, while a rarely used word appears near the root in balanced BSTs. Such an organization would slow down the translation, since the number of nodes visited when searching for a key in a BST depends on the depth of the node containing the key. We want words that occur frequently in the text to be placed nearer the root. (a) Design an efficient algorithm to minimize the number of nodes traversed in all searches in BST, given that we know how often each word occurs in a BST. What is the time complexity of your algorithm? (b) Given a set of 4 English words {Algorithm, Coding, Happy, Job offer}, corresponding French equivalent as {Algorithme, Codage, Heureuse, Offre d'emploi}, frequency of successful searches are (1, 4, 2, 1) and frequency of unsuccessful searches are (4, 2, 4, 1, 1). Apply your algorithm to determine the optimal cost of searching and structure of BST.	(4) (8)	CO2	L3


Bloom's Level wise Marks Distribution
0
6
8
9
22
Level 1
Level 2
Level 3
Level 4
Level 5
Level 6


Course Outcome wise Marks Distribution
32
28
24
20
16
12
8
4
0
18
27
CO1
CO2