INDEX

Name: **Enrollment No.:**

Branch & Batch: Artificial Intelligence & Data Science

Paper Title: Analysis and Design of Algorithm Lab **Paper Code:** ARM 254

Sheet No.:	Sub Part	Topic of Practical	Signature/ Remarks
LAB SHEET:1	**Q1**	Revise pseudocode for sorting an array (int, float, or char type) using following sorting techniques: ● Selection sort ● Bubble sort ● Merge sort (Recursive) ● Quick sort (Recursive) A. Plot the complexity chart for n=10 to 100. B. Analyze their complexities in best case, average case and worst case.	
	Q2	Revise pseudocode for searching within an array (int, float, or char type) using following searching techniques: ● Linear Search ● Binary Search A. Plot the complexity chart for n=10 to 100. B. Analyze their complexities in best case, average case and worst case.	
	Q3	You have been given two sorted lists of size M and N. It is desired to find the Kth smallest element out of M+N elements of both lists. Propose and implement an efficient algorithm to accomplish the task. Further, propose and implement an	

Name: **Enrollment No.:**

		efficient algorithm to accomplish the task considering that elements in both lists are unsorted.	
	Q4	You are given a list of n-1 integers and these integers are in the range of 1-n. There are no duplicates in the list. One of the integers is missing in the list. Write an efficient code to find the missing integer.	
	Q5	You have been given a sorted array ARR (of size M, where M is very large) of two elements, 0 and 1. It is desired to compute the count of 0s in the array ARR. Propose and implement an efficient algorithm to accomplish the task.	
LAB SHEET:2	**Q1**	Let there be an array of N random elements. We need to sort this array in ascending order. If n is very large (i.e. N= 1,00,000) then Quicksort may be considered as the fastest algorithm to sort this array. However, we can further optimise its performance by hybridising it with insertion sort. Therefore, if n is small (i.e. N<= 10) then we apply insertion sort to the array otherwise Quick Sort is applied. Implement the above discussed hybridised Quick Sort and compare the running time of normal Quick sort and hybridised quick sort. Run each type of sorting 10 times on a random set of inputs and compare the average time returned by these algorithms.	
LAB SHEET:3	**Q1**	Implement the strassen's multiplication method (using Divide and Conquer Strategy) and naive multiplication method. Compare these methods in terms of time taken using the nXn matrix where n=3, 4, 5, 6, 7 and 8 (compare in bar graph).	
	Q2	Implement the multiplication of two N-bit numbers (using Divide and Conquer Strategy) and naive multiplication method. Compare these methods in terms of time taken using N-bit numbers where n=4, 8, 16, 32 and 64.	
	Q3	Maximum Value Contiguous Subsequence: Given a sequence of n numbers A(1) ...A(n), give an algorithm for finding a contiguous subsequence A(i)...A(j) for which the sum of	

Name: **Enrollment No.:**

		elements in the subsequence is maximum. Example : {-2, 11, -4, 13, -5, 2} → 20 and {1,-3, 4, -2, -1, 6 } → 7.	
	Q4	Implement the algorithm (Algo_1) presented below and discuss which task this algorithm performs. Also, analyze the time complexity and space complexity of the given algorithm. Further, implement the algorithm with following modification: replace m = ⌈2n/3⌉ with m =⌊2n/3⌋, and compare the tasks performed by the given algorithm and modified algorithm.	
LAB SHEET:4	**Q1**	Implement LCS algorithm for A[1 .. n] and B[1 .. l] sequences.	
	Q2	Given an array A[1 .. n] of integers, compute the length of a longest increasing subsequence A sequence B[1 .. l] is increasing if B[i] > B[i − 1] for every index i ≥ 2. For example, given the array ⟨3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3, 2, 3, 8, 4, 6, 2, 7⟩	
	Q3	Given an array A[1 .. n] of integers, compute the length of a longest alternating subsequence. A sequence B[1 .. l] is alternating if B[i] < B[i − 1] for every even index i ≥ 2, and B[i] > B[i − 1] for every odd index i ≥ 3. For example, given the array ⟨3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3, 2, 3, 8, 4, 6, 2, 7⟩	
	Q4	Given an array A[1 .. n], compute the length of a longest palindrome subsequence of A. Recall that a sequence B[1 .. l] is a palindrome if B[i] = B[l− i + 1] for every index i.	
	Q5	Given an array A[1 .. n] of integers, compute the length of a longest convex subsequence of A. A sequence B[1 .. l] is convex if B[i] − B[i − 1] > B[i − 1]− B[i − 2] for every index i ≥ 3.	
LAB SHEET:5	**Q1**	Implement MCM algorithm for the given n matrix <M1xM2… ..... Mn> where the size of the matrix is Mi=di-1 x di.	
	Q2	Implement OBST for given n keys (K1,K2…..Km) whose pi and qi (dummy keys) are given.	

Name: **Enrollment No.:**

	Q3	Implement 0/1 Knapsack problem using dynamic programming.	
LAB SHEET:6	**Q1**	Wap to Implement breadth first search algorithm for given graph G.	
	Q2	Wap to Implement depth first search algorithm for given graph G.	
	Q3	Wap to Implement topological sorting.	
	Q4	Wap to find the strongly connected components in a Graph.	
LAB SHEET:7	**Q1**	Wap to Implement Prim's algorithm for given graph G.	
	Q2	Wap to Implement Kruskal's algorithm for given graph G.	
	Q3	Wap to Implement dijkstra algorithm to find single source shortest path.	