Swami Keshvanand Institute of Technology, Management &Gramothan, Jaipur

Department of Computer Science & Engineering

Assignment Sheet-I

Session 2024_25 (Odd Semester)

SET-XI

Q.No	Question	BL	CO	MM
1	The Subset Sum problem is to determine whether a subset of a set of numbers adds up to a given target sum. The brute-force solution has time complexity O(2^n). Design a more efficient dynamic programming algorithm that solves the problem in polynomial time. Evaluate your solution by comparing its time complexity with the brute-force approach.	5	1	10
2	You are given n activities with their start and end times. Select the maximum number of activities that a person can attend, assuming only one activity can be attended at a time. You must implement a greedy algorithm to solve this. Activities: 1 2 3 4 5 Start times: 1 3 0 5 8 End times: 2 4 6 7 9	4	1	10
3	Analyze **Binary Search**: • Show how the search space reduces by half in each step. • Derive the number of comparisons as a function of nnn and explain why it leads to O(logn). • Compare with Linear Search and justify why Binary Search is better for large datasets.	4	5	10

4	def max_subarray(arr):
	max_ending_here = max_so_far = arr[0]
	for x in arr[1:]:
	max_ending_here = max(x, max_ending_here + x)
	max_so_far = max(max_so_far, max_ending_here)
	return max_so_far
	Analyze the time complexity of this algorithm and compare it with the brute-force approach that checks every possible subarray. Justify why this solution is more efficient.
		4	5	10