मोतीलाल नेहरू राष्ट्रीय प्रौद्योगिकी संस्थान इलाहाबाद
प्रयागराज–211004 भारत

Motilal Nehru National Institute of Technology Allahabad
Prayagraj-211004 [India]

Computer Science and Engineering Department
Mid Semester (Even) Examination 2023-24

Programme Name: MCA | Semester: III

Course Code: CS33104

Course Name: Analysis of Algorithms | Student Reg. No.: 2 0 2 2 C A 0 2 3

Duration: 90 Minutes | Max. Marks: 25

Instructions: (Related to Questions)

1. *Precise answer with proper justification will be considered to get complete marks.*
2. *Attempt all questions.*

			Marks
Q1	a	Design an algorithm to rearrange elements of a given array of n real numbers so that all its negative elements precede all its positive elements without making use of an auxiliary array. Analyze the running time of your algorithm.	4
Q2	a	Suppose we have a set S $=\{a_1,a_2,\ldots\ldots a_n\}$ of n proposed activities that wish to use a resource, such as a lecture hall, which can serve only one activity at a time. Each activity a_i has a start time s_i and a finish time f_i, where $0\leq s_i\leq f_i\leq\infty$. If selected, activity a_i takes place during the half-open time interval $[s_i, f_i)$. Activities a_i and a_j are compatible if the intervals $[s_i, f_i)$ and $[s_j, f_j)$ do not overlap. That is, a_i and a_j are compatible if $s_i\geq f_j$ or $s_j\geq f_i$. In the activity-selection problem, we wish to select a maximum-size subset of mutually compatible activities. Design an optimal algorithm for the activity selection problem and prove your method and determine its complexity.	5
Q3	a	Given an array A, find the nonempty, contiguous subarray of A whose values have the largest sum and this contiguous subarray the maximum subarray. For example, in	

the array of Figure, the maximum subarray of A[1..16] is A[8 ...11], with the sum 43.

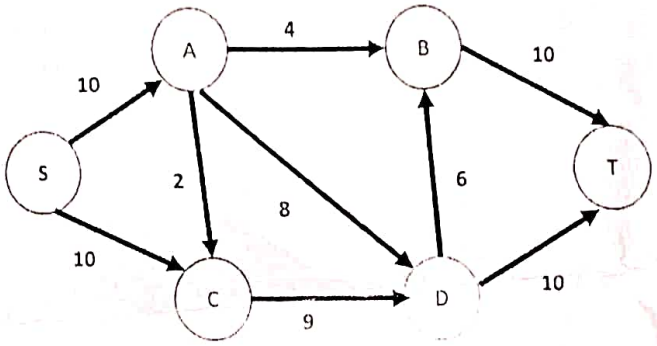
6

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	13	–3	–25	20	–3	–16	–23	18	20	–7	12	–5	–22	15	–4	7

maximum subarray

Computing the cost of one subarray might take time proportional to the length of the subarray, when computing all n^2 subarray sums leads to brute-force solution takes $O(n^2)$. Propose a better solution that you can solve this problem in O(nlogn) complexity.

Q 4 a Consider the following graph, G=(V, E) such that V≠0. Assume that 'S' is the source vertex. Determine the shortest path from source vertex to remaining vertices. Further, support your answer with the algorithm pseudo code with its time complexity. 5



Q 5 a Determine the complexity of this recurrence relation T(n)=T(n-1)+O(n) and prove it 2

b Dijkstra's algorithm doesn't work always. Justify the statement given supporting your answer with an example 2

c Justify under which instance when you apply quick sort, the array divides equally every time into half such that your time complexity will be O(nlogn). Support your answer with an example. 1