

MAHATMA GANDHI UNIVERSITY, KOTTAYAM
MGU-UGP (HONOURS)
MODEL QUESTION PAPER
MG5DSEMATA02:Design and Analysis of Algorithms
Fifth Semester

Duration: 2 Hours

Maximum Marks: 70

Students should attempt at least one question from each course outcome to enhance their overall attainability.

PART A

Answer any five questions. Each question carries 2 marks.

1. Define an algorithm. What are its essential characteristics? [CO1]
2. Distinguish between best-case and worst-case complexity. [CO2]
3. Define time complexity and space complexity. [CO2]
4. What is a greedy algorithm? [CO3]
5. What is meant by an optimal solution? [CO3]
6. What is meant by compatible intervals? [CO3]
7. Define the Weighted Interval Scheduling Problem. What is the objective of the problem? [CO4]
8. Define the SAT (Satisfiability) Problem and 3-SAT Problem. [CO5]

PART B

Answer any five questions. Each question carries 6 marks.

9. Prove that if

$$f(n) = O(g(n)) \quad \text{and} \quad g(n) = O(h(n)),$$

then

$$f(n) = O(h(n)).$$

[CO1]

10. Define an inversion in a schedule. Explain the role of inversions in proving the optimality of the Earliest Deadline First algorithm. [CO3]

11. Consider the following weighted intervals, already arranged in increasing order of finish time.

Interval j	1	2	3	4	5	6
Weight v_j	2	4	4	7	2	1
$p(j)$	0	0	1	0	3	3

Using the recurrence

$$M[j] = \max\{v_j + M[p(j)], M[j - 1]\},$$

compute

$$M[0], M[1], \dots, M[6]$$

and find the maximum total weight.

[CO4]

12. For $n = 4$ points, suppose the least-square errors are given by

$e_{i,j}$	$j = 1$	$j = 2$	$j = 3$	$j = 4$
$i = 1$	0	2	5	9
$i = 2$		0	1	4
$i = 3$			0	2
$i = 4$				0

If the penalty for each segment is $C = 3$, use

$$OPT(j) = \min_{1 \leq i \leq j} \{e_{i,j} + C + OPT(i - 1)\}$$

to calculate $OPT(1), OPT(2), OPT(3), OPT(4)$ and determine the minimum-penalty segmentation.

[CO4]

13. Given the array M , $p(j)$, and v_j for a weighted interval scheduling problem, use the Find-Solution procedure to trace back and identify the intervals in an optimal solution.
[CO4]
14. Explain the concept of polynomial-time reduction. What does $Y \leq_P X$ mean? [CO5]
15. Explain the black-box interpretation of polynomial-time reductions with a suitable example.
[CO6]
16. Explain how polynomial-time reductions can be used to compare the relative difficulty of computational problems. Give the example of reducing a new problem to a problem that already has a polynomial-time algorithm.
[CO6]

PART C

Answer any three questions. Each question carries 10 marks.

17. Explain Breadth-First Search (BFS) with algorithm, suitable example, BFS tree and running-time analysis. [CO1]
18. Given a graph, perform Depth-First Search (DFS) from a specified vertex, construct the DFS tree and analyze the running time. [CO2]
19. Define an inversion in a schedule. Explain the role of inversions in proving the optimality of the Earliest Deadline First algorithm. [CO3]
20. Explain Mergesort and derive its recurrence relation. [CO3]
21. Given
$$T(n) \leq 2T(n/2) + cn, \quad T(2) \leq c,$$
construct the recursion levels and determine the total running time. [CO4]
22. What is an NP-complete problem? Explain the two conditions that a problem must satisfy to be NP-complete. [CO5]