

Reg. No. :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Question Paper Code : 11171

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2026.

Fourth Semester

Computer Science and Engineering

CS 3401 – ALGORITHMS

(Regulations 2021)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — ($10 \times 2 = 20$ marks)

1. How does interpolation search differ from binary search in searching efficiency?
2. State the role of lower bounds in algorithm analysis.
3. What is the significance of transitive closure in directed graphs?
4. When does bi-connectivity is critical in network reliability?
5. State the Bellman principle of optimality.
6. How does dynamic programming improve memory usage over naive recursion?
7. How heuristic selection in state space search algorithms?
8. Why is solution space pruning critical in backtracking-based approaches?
9. What makes randomized algorithms powerful in cryptography applications?
10. How does approximation ratio affect algorithm suitability for bin packing?

PART B — ($5 \times 13 = 65$ marks)

11. (a) (i) Evaluate pattern search algorithms and propose improvements for large texts. (7)
(ii) Assess the computational cost of best case in linear search with examples. (6)

Or

- (b) (i) Write in detail about Rabin-Karp algorithm with an example. (6)
(ii) Write down the heap sort algorithm and explain with an example. (7)

12. (a) Explain in detail about Ford-Fulkerson algorithm for max flow problem.

Or

- (b) Illustrate in detail about DFS and BFS algorithms for graph traversal.

13. (a) Write in detail about dynamic programming and explain matrix chain multiplication problem.

Or

- (b) (i) Explore multi-stage graph problems beyond shortest path and provide solutions. (6)

- (ii) Explain greedy technique pitfalls with illustrative examples. (7)

14. (a) Implement the subset sum problem using pruning strategies and discuss improvements.

Or

- (b) (i) Investigate solution strategies for graph coloring. (6)

- (ii) Compare state-space and dynamic programming approaches for knapsack problems. (7)

15. (a) (i) Formulate a randomized primality testing algorithm for large prime numbers. (6)

- (ii) Benchmark randomized quick sort against standard quick sort on varying data sets. (7)

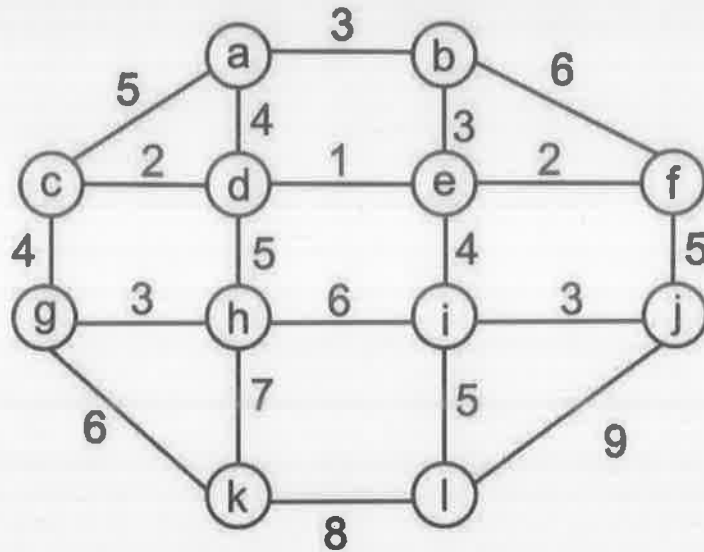
Or

- (b) (i) Compare and contrast between P, NP and NP-complete problems. (6)

- (ii) Illustrate application of randomized algorithms in big data analytics workflows. (7)

PART C — ($1 \times 15 = 15$ marks)

16. (a) Solve the following instance of single-source shortest paths problem with vertex a, as a source using Dijkstra's algorithm.



Or

- (b) Critically analyze the limits of polynomial-time approximation schemes (PTAS) for the Travelling Salesman Problem (TSP) in real-world routing domains. Design a scenario where standard PTAS fails to deliver acceptable performance either in terms of solution quality or computational resources. Develop and explain an improved algorithmic strategy to overcome these limitations while supporting your argument with complexity analysis and hypothetical experimental results.

