

2024

Time : 3 hours

Full Marks : 70

*Candidates are required to give their answers
in their own words as far as practicable.*

The figures in the margin indicate full marks.

Answer all sections as directed.

Section-A

Objective type Questions

Compulsory

1. Choose the correct answer of the following multiple choice questions : $2 \times 10 = 20$
 - (i) In the context of algorithm design, how do greedy algorithms differ from dynamic programming?
 - a. Dynamic programming is less efficient than greedy algorithms.

- b. Greedy algorithms always guarantee an optimal solution.
 - c. Greedy algorithms make local optimum choices without considering future consequences.
 - d. Dynamic programming solves sub-problems multiple times.
- (ii) How does time complexity impact the choice of algorithms in real-world applications?
- a. It is only relevant for theoretical analysis.
 - b. It determines the correctness of the algorithm.
 - c. It influences the algorithm's efficiency based on input size.
 - d. It has no effect on algorithm selection.
- (iii) What is the primary advantage of using dynamic programming over naive recursive methods?

- a. It simplifies the algorithm design process.
 - b. It solves each subproblem multiple times.
 - c. It stores results of sub-problems for future reference.
 - d. It guarantees a global optimum solution.
- (iv) In the context of algorithm design, what is a common misconception about greedy algorithms?
- a. They always produce the optimal solution.
 - b. They can be used for a wide range of problems.
 - c. They are less efficient than dynamic programming.
 - d. They require more memory than iterative techniques.
- (v) Which of the following best describes the iterative technique in algorithm design?

Greedy

a. A technique that builds solutions incrementally by making the best local choice at each step.

b. An approach that uses random sampling to find solutions.

Dynamic

c. A strategy that divides a problem into smaller subproblems and solves them independently.

Memorize

d. A method that solves problems through repeated application of a process until a condition is met.

(vi) What is a key advantage of using iterative techniques over recursive methods?

a. They require less memory than any other algorithm design technique.

b. They can solve any problem that recursive methods can.

c. They avoid the overhead of recursion, leading to better performance in certain cases.

d. They are always faster than recursive methods.

(vii) In what way does the Merge Sort algorithm exemplify the divide and conquer approach?

a. It sorts elements by repeatedly swapping adjacent items.

b. It divides the array into halves, sorts each half, and merges them back together.

c. It applies a greedy strategy to select the best elements at each step.

d. It uses a single loop to iterate through the array and sort it.

(viii) Which of the following statements accurately describes the difference between BFS and DFS?

a. Heap

b. Merge

c. Radix

d. None of these

- (ix) What misconception might someone have about the efficiency of sorting algorithms?
- Stable sorting algorithms are always slower than unstable ones.
 - All sorting algorithms can sort any type of data.
 - All sorting algorithms have the same time complexity.
 - Sorting algorithms are not important in computer science.
- (x) Which of the following best describes the process of Bubble Sort?
- It sorts elements by comparing their values in a binary tree structure.
 - It divides the array into halves and merges them back together.
 - It selects the smallest element from the unsorted region and moves it to the sorted region.

- d. It repeatedly compares adjacent elements and swaps them if they are in the wrong order.

Section-B

Short Answer type Questions

Answer any **four** questions of the following :

5×4=20

2. What is the significance of analyzing the correctness of an algorithm?
3. Explain the key differences between iterative and recursive techniques in algorithm design.
4. How does the Divide and Conquer approach work? Provide an example.
5. Write an algorithm for Linear Search.
6. What is the counting coin problem? Show with an example why the greedy algorithm is not very much effective in solving counting coins.
7. What is a Minimum Spanning Tree, and which algorithms are commonly used to find it?
8. Create a decision tree for a student registration process.

51

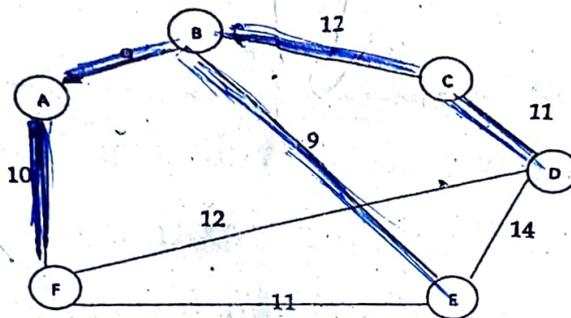
Section-C

Long Answer type Questions

Answer any two questions of the following :

$15 \times 2 = 30$

9. What is Radix Sort, and how does it differ from traditional comparison-based sorting algorithms?
10. How does Kruskal's algorithm work for finding the Minimum Spanning Tree? Use the diagram below to explain.



11. How can Breadth-First Search (BFS) be used to find the shortest path in an unweighted graph?
12. What are the key differences between Greedy Algorithms and Dynamic Programming?

S323/5/1

