

## Question Bank (UNIT -I)

1. Define an algorithm? Provide an example of a simple algorithm.
2. Define characteristic of an algorithm. Explain prior and posterior analysis.
3. What is the main purpose of analyzing an algorithm?
4. Defined time complexity in the context of algorithm analysis? What is the difference between **best-case**, **average-case**, and **worst-case** time complexity?
5. Define space complexity. Why is it important to analyze the space complexity of an algorithm?
6. How does space complexity differ from time complexity?
7. What does Big-O notation represent? Give an example.
8. Defined Big-O, Omega and Theta notation with proper example.
9. What is Omega ( $\Omega$ ) notation, and how does it differ from Big-O notation?
10. What does it mean when we say an algorithm has a time complexity of  $\Omega(n)$ ?
11. What is the divide and conquer method in algorithms? Can you give an example of an algorithm that uses this technique?
12. How does the divide and conquer approach differ from a greedy algorithm?
13. Describe the binary search algorithm. What is its time complexity in the worst case?
14. What is the time complexity of the binary search algorithm? Explain how the algorithm achieves this time complexity.
15. What is the best-case time complexity for binary search? When does this best case occur?
16. Describe the step-by-step process of how binary search works when searching for an element in a sorted array.
17. What is Merge Sort, and what is its time complexity?
18. Describe Merge Sort, and how does it work? Describe the general approach to the algorithm.
19. Write the algorithm for merge sort. Why does Merge Sort have a time complexity of  $O(n \log n)$  in the worst case.
20. What is Quick Sort, and how does it differ from Merge Sort in terms of sorting approach? What is the worst-case time complexity of Quick Sort?
21. What is the time complexity of Quick Sort in the best, average, and worst-case scenarios? Why does Quick Sort sometimes perform poorly in the worst case?

22. Write down the algorithm for Quick Sort. What is the time complexity for best case worst case and average case for quick sort?
23. What is Strassen's Matrix Multiplication, and how does it improve over traditional matrix multiplication?
24. Write down the algorithm for Strassen's Matrix Multiplication. What is the time complexity for best case worst case and average case for Strassen's Matrix Multiplication?
25. What is the time complexity of Strassen's Matrix Multiplication compared to traditional matrix multiplication?