

DAA Oral Questions

1. What is time complexity and Space complexity?
2. What is an asymptotic notation?
3. What are different types of asymptotic notations?
4. What is Big-O Notation?
5. What is time and space complexity?
6. What is meant by Amortized analysis?
7. What are different strategies of algorithms?
8. Explain divide and conquer strategies?
9. Explain Greedy method? Give examples?
10. What is difference between Greedy algorithm and Divide and Conquer?
11. What are P-class and NP- class problems?
12. What is NP hard and NP complete?
13. What is Big O notation?
14. What is time complexity of Quick sort, Merge sort?
15. What is Best, worst and average case
16. What is Computational complexity theory
17. What is knapsack problem?
18. What is vertex cover problem?
19. What is time complexity of binary search, linear search?
20. What is meant by job scheduling with deadline?
21. Time complexity of heap sort, merge sort and quick sort?
22. What is PRAM?
23. What is OBST?
24. What is parallel programming model?
25. Explain knapsack using different approach?
26. What is backtracking?
27. What is dynamic programming model?
28. What is Hamiltonian cycle?
29. What is traveling salesman problem?
30. What are 8 queen problems?
31. What is graph theory?
32. What is multistage graph?
33. What is difference between parallel and concurrent algorithms?
34. What is randomized algorithm
35. what is approximation algorithm?
36. What is IOT?
37. What are different distributed algorithms?

DAA Gate questions

Gate questions on Design and Analysis of Algorithms

Question 1

What is recurrence for worst case of QuickSort and what is the time complexity in Worst case?

- A Recurrence is $T(n) = T(n-2) + O(n)$ and time complexity is $O(n^2)$
- B Recurrence is $T(n) = T(n-1) + O(n)$ and time complexity is $O(n^2)$
- C Recurrence is $T(n) = 2T(n/2) + O(n)$ and time complexity is $O(n \log n)$
- D Recurrence is $T(n) = T(n/10) + T(9n/10) + O(n)$ and time complexity is $O(n \log n)$

Question 2

Suppose we have a $O(n)$ time algorithm that finds median of an unsorted array. Now consider a QuickSort implementation where we first find median using the above algorithm, then use median as pivot. What will be the worst case time complexity of this modified QuickSort.

- A $O(n^2 \log n)$
- B $O(n^2)$
- C $O(n \log n \log n)$
- D $O(n \log n)$

Question 3

Given an unsorted array. The array has this property that every element in array is at most k distance from its position in sorted array where k is a positive integer smaller than size of array. Which sorting algorithm can be easily modified for sorting this array and what is the obtainable time complexity?

- A Insertion Sort with time complexity $O(kn)$
- B Heap Sort with time complexity $O(n \log k)$
- C Quick Sort with time complexity $O(k \log k)$
- D Merge Sort with time complexity $O(k \log k)$

Question 4

Which of the following is not true about comparison based sorting algorithms?

- A The minimum possible time complexity of a comparison based sorting algorithm is $O(n \log n)$ for a random input array
- B Any comparison based sorting algorithm can be made stable by using position as a criteria when two elements are compared
- C Counting Sort is not a comparison based sorting algorithm
- D Heap Sort is not a comparison based sorting algorithm.

Question 5

What is time complexity of fun()?

```
int fun(int n)
{
    int count = 0;
    for (int i = n; i > 0; i /= 2)
        for (int j = 0; j < i; j++)
            count += 1;
    return count;
}
```

- A $O(n^2)$
- B $O(n \log n)$
- C $O(n)$
- D $O(n \log n \log n)$

Question 6

What is the time complexity of fun()?

```
int fun(int n)
{
    int count = 0;
    for (int i = 0; i < n; i++)
        for (int j = i; j > 0; j--)
            count = count + 1;
    return count;
}
```

- A $\Theta(n)$
- B $\Theta(n^2)$
- C $\Theta(n \log n)$
- D $\Theta(n \log n \log n)$

Question 7

The recurrence relation capturing the optimal time of the Tower of Hanoi problem with n discs is. (GATE CS 2012)

- A $T(n) = 2T(n - 2) + 2$
- B $T(n) = 2T(n - 1) + n$
- C $T(n) = 2T(n/2) + 1$
- D $T(n) = 2T(n - 1) + 1$

Question 9

Which of the following is not $O(n^2)$?

- A $(15^{10}) * n + 12099$
- B $n^{1.98}$
- C $n^3 / (\text{sqrt}(n))$
- D $(2^{20}) * n$

Question 12

What is the best time complexity of bubble sort?

- A N^2
- B $N \log N$
- C N
- D $N(\log N)^2$

Question 13

What is the worst case time complexity of insertion sort where position of the data to be inserted is calculated using binary search?

- A N
- B $N \log N$
- C N^2
- D $N(\log N)^2$

Question 14

The tightest lower bound on the number of comparisons, in the worst case, for comparison-based sorting is of the order of

- A N
- B N^2
- C $N \log N$
- D $N(\log N)^2$

Question 15

In a modified merge sort, the input array is splitted at a position one-third of the length(N) of the array. What is the worst case time complexity of this merge sort?

- A $N(\log N \text{ base } 3)$
- B $N(\log N \text{ base } 2/3)$
- C $N(\log N \text{ base } 1/3)$
- D $N(\log N \text{ base } 3/2)$

Question 16

What is the time complexity of the below function?

```

void fun(int n, int arr[])
{
    int i = 0, j = 0;
    for(; i < n; ++i)
        while(j < n && arr[i] < arr[j])
            j++;
}

```

- A $O(n)$
- B $O(n^2)$
- C $O(n \log n)$
- D $O(n(\log n)^2)$

Question 17

In a competition, four different functions are observed. All the functions use a single for loop and within the for loop, same set of statements are executed. Consider the following for loops:

- A) for($i = 0; i < n; i++$)
- B) for($i = 0; i < n; i += 2$)
- C) for($i = 1; i < n; i *= 2$)
- D) for($i = n; i > -1; i /= 2$)

If n is the size of input(positive), which function is most efficient(if the task to be performed is not an issue)?

- A A
- B B
- C C
- D D

Question 19

What does it mean when we say that an algorithm X is asymptotically more efficient than Y ?

- A X will be a better choice for all inputs
- B X will be a better choice for all inputs except small inputs
- C X will be a better choice for all inputs except large inputs
- D Y will be a better choice for small inputs

Question 20

What is the time complexity of Floyd-Warshall algorithm to calculate all pair shortest path in a graph with n vertices?

- A $O(n^2 \log n)$
- B $\Theta(n^2 \log n)$

C $\Theta(n^4)$

D $\Theta(n^3)$

Question 23

Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let $T(n)$ be the number of comparisons required to sort n elements. Then

A $T(n) \leq 2T(n/5) + n$

B $T(n) \leq T(n/5) + T(4n/5) + n$

C $T(n) \leq 2T(4n/5) + n$

D $T(n) \leq 2T(n/2) + n$

Question 26

Which of the following sorting algorithms has the lowest worst-case complexity?

A Merge Sort

B Bubble Sort

C Quick Sort

D Selection Sort