

DAA Interview Questions and Answers

A list of top frequently asked **DAA Interview Questions** and answers are given below.

1) What is Algorithm?

The name 'Algorithm' refers to the sequence of instruction that must be followed to clarify a problem.

The logical description of the instructions which may be executed to perform an essential function.

Algorithms are usually generated **independent of primary languages**, i.e., an algorithm can be achieved in more than one programming language.

[Read More](#)

2) What is Asymptotic Notation?

A way to represents the behavior of functions in the limit or without bounds.

The notations are described in terms of methods whose domains are the set of natural numbers $N = \{0, 1, 2\}$

Such notations are convenient for defining the worst-case running time function $T(n)$.

It can also be extended to the domain of the real number.

[Read More](#)

3) What is the time complexity of Algorithm?

The time complexity of an algorithm denoted the total time needed by the program to run to completion. It is generally expressed by using the **big O** notation.

4) Explain the algorithms for Bubble sort and give a suitable example.

In bubble sort technique the list is split into two sub-lists sorted and unsorted. The smallest component is bubbled from unsorted sub-list. After moving the smallest component, the imaginary wall moves one element ahead. The bubble sort was initially written to bubble up the highest item in the list. But there is no difference whether the highest / lowest item is bubbled. This technique is simple to understand but time-consuming. In this type, two successive components are compared, and swapping is done. Thus, step-by-step entire array items are checked, given a list of 'n' elements the bubble sort needed up to n-1 passes to sort the record.

[Read More](#)

5) Explain the algorithm for selection sort and give a suitable example.

In selection sort, the list is split into two sub-lists sorted and unsorted. These two lists are split by an imaginary wall. We find the smallest item from unsorted sub-list and swap it to the starting. And the wall moves one item ahead, as the sorted file is increases and an unsorted file is decreased.

Assume that we have a file on n elements. By applying a selection sort, the first item is compared with all remaining (n-1) elements. The smallest item is placed at the first location. Again, the second item is compared with the remaining (n-1) elements. At the time of the comparison, the smaller item is swapped with a bigger item. Similarly, the entire array is checked for smallest component, and then swapping is done accordingly. Here we need n-1 passes or repetition to rearrange the data completely.

[Read More](#)

6) Explain the algorithms for QUICK sort (partition exchange sort) and give a suitable example.

Quicksort is based on division. It is also called a partition exchange sorting. The basic concept of quick sort method is to pick one item from an array and rearranges the remaining item around it. This element split the main list into two sublists. This chosen item is known as a pivot. Once the pivot is selected, then it shifts all the components less than pivot to the left of value pivot, and all the items higher than the pivot are shifted to the right side. This process of choosing pivot and division the list is tested recursively until sub-lists consisting of only one element.

[Read More](#)

7) What is NP-Complete?

An NP-Complete problem is a problem in NP that is as difficult as any other trouble in this class because any other dispute in NP can be decreased to it in Polynomial-time.

8) Differentiate Time Efficiency and Space Efficiency?

Time Efficiency measured by estimate the number of times the essential algorithms functions are executed. Space Efficiency is measured by calculating the number of additional memory units consumed by the algorithm.

9) What is the Order of Algorithm?

The order of algorithm is standard documentation of an algorithm that has been developed to represent a task that bound the estimated time for algorithms. The order of an algorithm is a method of defining its efficiency. It is commonly referred to as O-notation.

10) What is Brute Force?

Brute Force is a straightforward method for solving problem, usually directly based on the problem's statement and descriptions of the concepts involved.

11) What are the various criteria used to improve the effectiveness of the algorithm?

Input- Zero or more quantities are externally provided.

Output- At least one quantity is composed

Definiteness- Each instruction is simple and unambiguous

Finiteness- If we trace out the instructions of an algorithm, then for all step the algorithm complete after a finite number of steps

Effectiveness- Every instruction must be elementary.

12) Explain the algorithms for Merge sort and give a suitable example.

The basic concept of merge sort has split the list into two smaller sub-lists of relatively equal size. Recursively repeat this method until only one item is left in the sub-list. After this, various sorted sub-lists are combined to form a sorted parent list. This method goes on recursively till the original sorted list arrived.

[Read More](#)

13) What is a Linear Search?

Linear search method is also called a sequential search technique. The linear search is a technique of searching an item in a list in sequence. In this technique, the array is searched for the required item from the starting of the list/array or the last component to the first component of the array and continues until the element is found or the entire list/array has been searched.

Advantages

1. It is an easy and conventional technique of searching for information. The linear or sequential name implies which the elements are stored in a systematic manner.
2. The item in the list can be in any order. i.e., the linear search can be tested on the sorted or unsorted linear data structure.

Disadvantages

1. This procedure is insufficient when a large number of item is present in the list.
2. It consumes more time and decreases the retrieval rate of the system.

Time complexity: $O(n)$

14) What is Binary Search?

Binary search is higher than the linear search. However, it cannot be tested on the unsorted data structure. The binary search is based on the method divide-and-conquer. The binary search begins by testing the data in the middle component of the array. This determines an object is whether in the first half or second half. If the object is in the first half, we do not need to check the second half and if it is in the second half no need to check in first half. Similarly, we repeat this procedure until we find an object in the list or not found from the list. Here we need three variables to identify first, last, and middle elements.

To implement a binary search technique, the item must be in sorted order.

Binary Search is performed as follows:

- The key is compared with an element in the middle position of an array
- If the key matches with an element, return it, and stop.
- If the key is less than mid position element, then the element to be found must be in the first half of the array, otherwise it must be in the second half of the array.
- Repeat the method for lower (or upper half) of the array until the component is found.

[Read More](#)

15) Explain the algorithms for insertion sort and give a suitable example.

Both the selection and bubble sorts exchange items. But insertion sort does not exchange items. In insertion sort, the item is inserted at an appropriate place similar to card insertion. Here the list is split into two parts sorted and unsorted sub-lists. In each pass, the first component of unsorted sublist is picked up and moved into the sorted sublist by inserting it in a relevant position. Suppose we have '**n**' items, we need **n-1** passes to sort the items.

[Read More](#)

16) What do you mean by the optimal solution?

Given the problem with inputs, we recover a subset that appeases come constraints. Any subset that satisfies these constraints is known as a feasible solution. A feasible solution, which either maximizes or minimizes a given purpose method is known as an optimal solution.

17) Why Hashing?

Suppose we have a huge information set stored in an array. The amount of time needed to lookup an item in the array is either **$O(\log n)$** or **$O(n)$** based on whether the array is sorted or not. If the array is sorted, then a procedure such as a binary search can be used to search the array. Otherwise, the array must be searched linearly. Either method may not be desirable if we need to process a very high data set. Therefore we discuss a new procedure called **hashing** that allows us to update and fetch any entry in constant time **$O(1)$** . The constant time or **$O(1)$** performance defines the amount of time to operate does not depend on data size n .

[Read More](#)

18) Explain how the encryption algorithm works?

Encryption is the step of converting plaintext into a secret code format called "**Ciphertext**". To convert the content, the algorithm uses a string of bits referred to as "**keys**" for estimation. The larger the key, the higher the number of potential patterns for generating the ciphertext. Most encryption algorithm use codes fixed blocks of input that have a length of about 64 to 128 bits, while some uses stream technique.

19) What is Dynamic Programming?

DP is another method for problems with optimal substructure: An optimal solution to a problem include optimal solutions to subproblems. This doesn't necessarily define that every optimal solution to a subproblem will contribute to the primary solution.

For **divide and conquer** (top-down), the subproblems are independent, so we can resolve them in any order.

For the **greedy technique** (bottom-up), we can always choose the "right" subproblem by a greedy choice.

In **dynamic programming**, we solve many subproblems and save the results: not all of them will contribute to solving the bigger problem. Because of optimal substructure, we can be sure that at least some of the subproblems will be useful.

[Read More](#)

20) What is the Knapsack Problem?

Given n elements of known weights w_i and values v_i , $i=1, 2, \dots, n$, and a knapsack of capacity W , find the most valuable subsets of the elements that fit the knapsack. It is convenient to order the elements of a given instance in descending order by their value-to-weight ratios. Then the first element gives the best payoff per weight unit, and the last one gives the worst payoff per weight unit.

[Read More](#)

21) What is Warshall's Algorithm?

Warshall's algorithm is a function of dynamic programming procedure, which is used to find the transitive closure of a directed graph.

22) What is a Greedy Algorithm?

A greedy technique for an optimization problem always makes the option which look best at the moment and add it to the current subsolution.

[Read More](#)

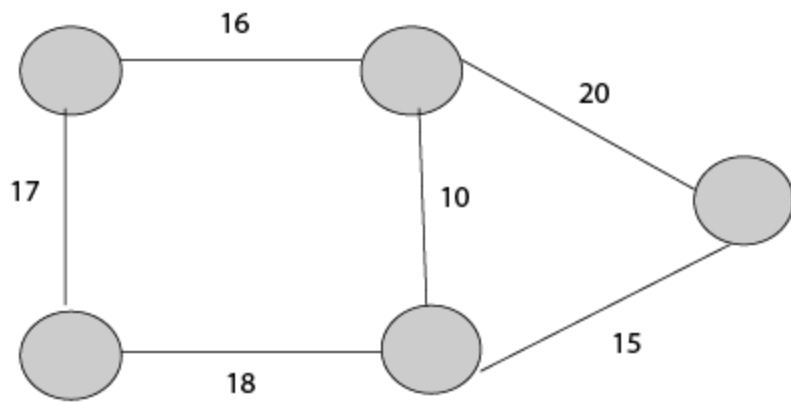
23) List the advantage of the greedy algorithm.

1. The greedy method produces a feasible solution
 2. The greedy method is very easy to solve a problem
 3. The greedy method implements an optimal solution directly
-

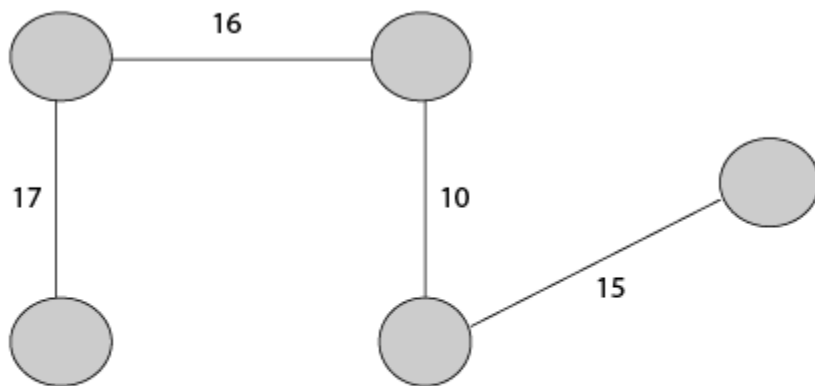
24) What is Minimum Spanning Trees?

A spanning tree for a linked graph is a tree whose vertex set is the same as the vertex set of the given graph, and whose edge set is a subgroup of the edge set of the given graph. i.e., any linked graph will have a spanning tree.

Weight of a spanning tree $w(T)$ is the total of weights of all edges in T . The Minimum spanning tree (MST) is a spanning tree with the smallest feasible weight.



Connected , Undirected Graph



Minimum Cost Spanning Tree

Total Cost = $17+16+10+15=58$

[Read More](#)

25) What is Kruskal's Algorithm?

This is a greedy method. A greedy method chooses some local optimum (i.e., selection an edge with the smallest weight in an MST).

Kruskal's algorithm works as follows:

Take a graph with 'n' vertices, keep on adding the shortest (least cost) edge, while avoiding the generation of cycles, until $(n - 1)$ edges have been added. Frequently two or more edges may have the same rate. The order in which the edges are decided, in this method, does not matter. Different Minimum spanning tree may result, but they will all have the same total price, which will always be the minimum cost.

[Read More](#)

26) What is Sorting Network?

A sorting network is a numerical model of a network of wires and comparator modules that is used to sort a series of numbers. Each comparator connects two wires and sorts the values by outputting the smaller value to one wire, and the higher to the other. The main difference between **sorting networks** and **comparison sorting** algorithms is that with a sorting network, the series of comparisons is set in advance, regardless of the result of previous comparisons. This independence of comparison series is useful for parallel execution of the methods.

There are three types of Sorting Network



[Read More](#)

27) What is Floyd's algorithm?

Floyd's algorithm is a function, which is used to find all the pairs shortest paths problem. Floyd's algorithm is relevant to both directed and undirected weighted graph, but they do not include a cycle of a negative length.

28) What is prim's algorithm?

Prim's algorithm is a greedy and efficient technique, which is used to find the minimum spanning the tree of a weighted linked graph.

29) How efficient is prim's algorithm?

The efficiency of the prim's methods depends on the data structure chosen for the graph.

30) What is Dijkstra's Algorithm?

Dijkstra's algorithm solves the single-source shortest path method of finding shortest paths from a given vertex (the source), to another vertex of a weighted graph or digraph. Dijkstra's algorithm implements a correct solution for a graph with non-negative weights.

[Read More](#)

31) What are the huffman trees?

A Huffman tree is a binary tree which reduces the weighted path length from the root to the leaves, including a set of predefined weights. The essential application of Huffman trees is a Huffman code.

32) What do you mean by Huffman code?

A Huffman code is an optimal prefix tree variable-length encoding technique which assign bit strings to characters based on their frequency in a given text.

[Read More](#)

33) List the advantage of Huffman's encoding?

Huffman's encoding is one of the essential file compression techniques.

1. It is easy
2. It is flexibility
3. It implements optimal and minimum length encoding

34) What is dynamic Huffman coding?

In dynamic Huffman coding, the coding tree is updated each time a new character is read from the source text. Dynamic Huffman n-coding used to overcome the disadvantage of the simplest version.

35) What is backtracking?

Depth-first node generation with bounding method is known as backtracking. The backtracking technique has its virtue the ability to yield the solution with far fewer than m trials.

[Read More](#)

36) Write the difference between the Dynamic programming and Greedy method.

Dynamic programming

1. Many numbers of decisions are generated.
2. It gives an optimal solution always

Greedy method

1. Only one sequence of decision is generated.
2. It does not require to provide an optimal solution always.

[Read More](#)

37) What is the use of Dijkstra's algorithm?

Dijkstra's procedure is used to solve the single-source shortest-paths method: for a given vertex called the source in a weighted linked graph, find the shortest path to all its other vertices. The single-source shortest-paths process asks for a family of paths, each leading from the source to various vertex in the graph, though some direction may have edges in common.

38) What is meant by n-queen Problem?

The problem is to area n-queens on an n-by-n chessboard so that no two queens charge each other by being same row or in the same column or the same diagonal.

39) What is the state-space tree?

The processing of backtracking is resolved by constructing a tree of choices being made. This is known as state-space tree. Its root describes an initial state before the search for a solution starts. The nodes of the first level in the tree describe the choices made for the first element of the solution, the nodes in the second level describe the choices for the second element, and so on.

40) What is the assignment problem?

Assigning 'n' people to 'n' jobs so that the total price of the assignment is as low as possible. The instance of the problem is particularized as an n-by-n cost matrix C so that the problem can be described as select one element in each row of the matrix so that no two selected items are in the same column and the total is the smallest possible.
