DAA(Data Analysis and Approach)

Q1 Which of the following algorithms is not based on a divide-and-conquer approach?

a) QuickSort

b) Binary Search

c) Linear Search

d) MergeSort

Ans: Linear Search

EXP: Linear Search does not use a divide-and-conquer approach. Instead, it sequentially

checks each element until it finds the target or reaches the end of the list.

Q2 What is the best-case time complexity of bubble sort?

a) *O*(1)

b) *O*(*n*)

c) *O*(*n* log *n*)

d) *O*(*n*^2)

Ans: *O*(1)

Exp: In the best-case scenario, Bubble Sort can detect that the list is already sorted after just

one pass, making its best-case time complexity �(�)*O*(*n*).

Q3 An algorithm that requires 2n operations to complete is said to have what kind of time

complexity?

a) Linear

b) Logarithmic

c) Polynomial

d) Exponential

Ans Exponential

Exp: An algorithm that requires 2noperations is said to have exponential time complexity.

Exponential time complexities often arise in algorithms that process all combinations of the

input.

Q4: Which of the following algorithmic paradigms solves problems by trying out solutions

likely to succeed?

a) Dynamic Programming

b) Backtracking

c) Greedy

d) Divide and Conquer

Ans: Greedy

Exp: The Greedy algorithm makes the most promising choice at each decision point. It hopes

that making a series of locally optimal solutions will lead to a globally optimal solution.

Q5: Which of the following is an in-place sorting algorithm?

a) MergeSort

b) QuickSort

c) HeapSort

d) Counting Sort

Ans: QuickSort

Exp: In-place sorting algorithms use a constant amount of extra space for producing the

output. QuickSort sorts the elements by swapping them in place, without needing additional

storage. MergeSort, on the other hand, requires additional space proportional to the input

size.

Q6: Which of the following data structures is used by the breadth-first search (BFS)

algorithm?

a) Stack

b) Queue

c) Linked List

d) Tree

Ans: Queue

Exp: BFS (Breadth-First Search) uses a queue to keep track of nodes that it needs to explore.

It processes nodes level by level, ensuring that nodes are visited in increasing order of their

distance from the starting node.

Q7: The quicksort algorithm is also known as:

a) Partition-exchange sort

b) Divide-exchange sort

c) Partition-swap sort

d) Divide-swap sort

Ans: Partition Exchange Sort

Exp: Quicksort is also known as "Partition-exchange sort" because it works by selecting a

'pivot' element from the array and partitioning the other elements into two sub-arrays,

according to whether they are less than or greater than the pivot.

Q8: Which of the following problems can be solved using the dynamic programming approach?

a) Traveling Salesman Problem

b) Maximum Subarray Sum

c) Graph Coloring

d) Tower of Hanoi

Ans: Maximum Subarray Sum

Exp: The maximum subarray sum problem can be efficiently solved using dynamic programming.

The Tower of Hanoi is a recursive problem, and the Traveling Salesman Problem and Graph

Coloring are NP-hard problems that do not have efficient solutions using dynamic programming.

Q9: Which of the following algorithms is used to find the shortest path in a weighted graph?

a) Bellman-Ford algorithm

b) Breadth-First Search (BFS)

c) Depth-First Search (DFS)

d) Kruskal's algorithm

Ans: Bellman-Ford algorithm

Exp: The Bellman-Ford algorithm computes shortest paths from a single source vertex to all of the

other vertices in a weighted graph. While BFS can determine shortest paths, it's for unweighted

graphs.

Q10: What is the worst-case time complexity of the binary search algorithm?

*Ans: O*(log*n*)

Q11: Which of the following algorithms works by repeatedly finding the smallest element from the

unsorted portion and swapping it with the first unsorted element?

a) Insertion Sort

b) Bubble Sort

c) Selection Sort

d) Merge Sort

Ans: Selection Sort

Exp: The Selection Sort works by repeatedly finding the smallest (or largest, depending on sorting

order) element from the unsorted part and swapping it with the first unsorted element.

Q12: In which sorting algorithm are elements continuously swapped until no more swaps are

needed?

a) Insertion Sort

b) Bubble Sort

c) QuickSort

d) Selection Sort

Ans: Bubble Sort

Exp: Bubble Sort repeatedly steps through the list, compares adjacent elements, and swaps them if

they are in the wrong order. This process is repeated until no swaps are needed, indicating that the

list is sorted.

Q13: The time complexity of an algorithm that runs in constant time does not depend on the size of the input data. This is represented as:

Ans: O(1)

Exp: Constant time complexity, represented as O(1), means that the running time of the algorithm

does not depend on the size of the input data.

Q14: Which of the following algorithms is particularly useful when the list is already mostly sorted?

a) Bubble Sort

b) Insertion Sort

c) Merge Sort

d) QuickSort

Ans: Insertion Sort

Exp: Insertion Sort is adaptive, meaning it becomes faster when dealing with a mostly sorted list. If

only a few elements are out of order, insertion sort can handle them with a minimal number of

comparisons and swaps.

Q15: An algorithm that sometimes makes a choice which looks best at the moment without

considering the consequences is called:

a) Dynamic Programming

b) Backtracking

c) Greedy

d) Divide and Conquer

Ans: Greedy

Exp: A Greedy algorithm makes the most promising choice at each decision point. It looks for the

best option at the current moment without worrying about the consequences of that choice