## CS502 Fundamentals of Algorithms

## Important mcqs

#### Lec 1 - Introduction

### 1. What is the purpose of an introduction?

- a) To summarize the entire work
- b) To establish the author's credibility
- c) To provide supporting evidence
- d) To present the conclusion

Answer: b) To establish the author's credibility

#### What should an introduction include?

- a) The author's personal opinions
- b) A thesis statement
- c) Background information unrelated to the topic
- d) A conclusion

Answer: b) A thesis statement

#### Why is the introduction considered a critical component of any communication?

- a) It sets the tone for the entire piece
- b) It summarizes the main points of the work
- c) It provides supporting evidence
- d) It presents a conclusion

Answer: a) It sets the tone for the entire piece

#### What does a well-crafted introduction do?

- a) Confuses the reader
- b) Bore the reader
- c) Captures the audience's attention
- d) Presents irrelevant information

Answer: c) Captures the audience's attention

### Which of the following is NOT included in an introduction?

- a) A thesis statement
- b) A summary of the entire work
- c) Background information related to the topic
- d) The author's personal opinions

Answer: b) A summary of the entire work

#### Which part of a piece of writing or speech is the introduction?

- a) The conclusion
- b) The middle
- c) The beginning

d) It can be anywhere in the piece

Answer: c) The beginning

#### What is the main goal of an introduction?

- a) To provide evidence to support the author's argument
- b) To persuade the reader to agree with the author's opinion
- c) To establish the author's credibility and interest the reader
- d) To present a conclusion

Answer: c) To establish the author's credibility and interest the reader

#### How can an introduction be effective?

- a) By providing irrelevant information
- b) By including a thesis statement and relevant background information
- c) By presenting a conclusion
- d) By confusing the reader

Answer: b) By including a thesis statement and relevant background information

#### Which of the following is NOT a purpose of an introduction?

- a) To establish the author's credibility
- b) To provide supporting evidence
- c) To motivate the reader to continue reading or listening
- d) To present a thesis statement

Answer: b) To provide supporting evidence

#### What does the introduction serve as?

- a) A conclusion
- b) A guide to the reader or listener
- c) An explanation of the author's personal opinions
- d) A summary of the entire work

Answer: b) A guide to the reader or listener

### **Lec 2 - Asymptotic Notation**

- 1. What is the purpose of using asymptotic notation in the analysis of algorithms?
  - a) To focus on the growth rate of the function
  - b) To consider constant factors and lower order terms
  - c) To provide precise measurements of algorithm efficiency
  - d) To identify the fastest algorithm

Answer: a) To focus on the growth rate of the function

#### Which asymptotic notation represents the upper bound of a function?

- a) Big O
- b) Omega
- c) Theta
- d) None of the above

Answer: a) Big O

### Which asymptotic notation represents the lower bound of a function?

- a) Big O
- b) Omega
- c) Theta
- d) None of the above

Answer: b) Omega

### Which asymptotic notation represents both the upper and lower bounds of a function?

- a) Big O
- b) Omega
- c) Theta
- d) None of the above

Answer: c) Theta

#### Which of the following statements is true about Big O notation?

- a) It represents the exact running time of an algorithm
- b) It represents the best-case running time of an algorithm
- c) It represents the worst-case running time of an algorithm
- d) It represents the average-case running time of an algorithm

Answer: c) It represents the worst-case running time of an algorithm

## Which of the following notations is used to describe the best-case running time of an algorithm?

- a) Big O
- b) Omega
- c) Theta
- d) None of the above

Answer: b) Omega

## Which of the following notations is used to describe the average-case running time of an algorithm?

- a) Big O
- b) Omega
- c) Theta

d) None of the above

Answer: d) None of the above (average-case running time is typically not described using asymptotic notation)

Which of the following functions has a higher growth rate:  $f(n) = n^2$  or  $g(n) = 2^n$ ?

- a)  $f(n) = n^2$
- b)  $g(n) = 2^n$
- c) They have the same growth rate

Answer: b)  $g(n) = 2^n$ 

Which of the following functions has a lower growth rate:  $h(n) = \log n$  or j(n) = n?

- a)  $h(n) = \log n$
- b) j(n) = n
- c) They have the same growth rate

Answer: a) h(n) = log n

Which of the following notations can be used to describe an algorithm with a constant running time?

- a) Big O
- b) Omega
- c) Theta
- d) None of the above

Answer: c) Theta (since constant time is both an upper and lower bound)

### Lec 3 - Divide and Conquer Strategy

#### 1. What is the purpose of the Divide and Conquer strategy?

- a) To break down a complex problem into smaller subproblems.
- b) To solve a problem recursively.
- c) To combine the solutions of smaller subproblems to obtain the final solution.
- d) All of the above.

Answer: d) All of the above.

#### Which of the following problems can be solved using the Divide and Conquer strategy?

- a) Sorting an array of integers.
- b) Finding the shortest path between two points in a graph.
- c) Calculating the value of an arithmetic expression.
- d) All of the above.

Answer: d) All of the above.

### What is the time complexity of the Divide and Conquer strategy?

- a) O(n)
- b) O(log n)
- c) O(n log n)
- d) O(n^2)

Answer: c) O(n log n)

#### Which of the following is not a step involved in the Divide and Conquer strategy?

- a) Breaking down the problem into smaller subproblems.
- b) Solving the subproblems recursively.
- c) Combining the solutions of smaller subproblems.
- d) None of the above.

Answer: d) None of the above.

### Which of the following is an example of the Divide and Conquer strategy?

- a) Merge sort.
- b) Quick sort.
- c) Binary search.
- d) All of the above.

Answer: d) All of the above.

#### Which of the following is true about the Divide and Conquer strategy?

- a) It is a top-down approach.
- b) It is a bottom-up approach.
- c) It can be both top-down and bottom-up.
- d) None of the above.

Answer: a) It is a top-down approach.

#### What is the main advantage of the Divide and Conquer strategy?

- a) It simplifies complex problems.
- b) It is easy to implement.
- c) It has a fast running time.
- d) None of the above.

Answer: c) It has a fast running time.

Which of the following problems cannot be solved using the Divide and Conquer

#### strategy?

- a) Multiplying two large integers.
- b) Finding the maximum element in an array.
- c) Calculating the Fibonacci sequence.
- d) All of the above can be solved using the Divide and Conquer strategy.

Answer: b) Finding the maximum element in an array.

### Which sorting algorithm uses the Divide and Conquer strategy?

- a) Bubble sort.
- b) Insertion sort.
- c) Merge sort.
- d) Selection sort.

Answer: c) Merge sort.

## Which of the following is true about the subproblems generated in the Divide and Conquer strategy?

- a) They must be of equal size.
- b) They must be disjoint.
- c) They can be of different sizes.
- d) None of the above.

Answer: c) They can be of different sizes.

### Lec 4 - Sorting

- 1. Which of the following is not a sorting algorithm?
  - a) Merge sort
  - b) Bubble sort
  - c) Hash sort
  - d) Quick sort

Solution: c) Hash sort

### Which sorting algorithm has the worst-case time complexity of O(n^2)?

- a) Quick sort
- b) Merge sort
- c) Bubble sort
- d) Radix sort

Solution: c) Bubble sort

## Which of the following sorting algorithms is a stable sort?

- a) Heap sort
- b) Insertion sort
- c) Quick sort
- d) Selection sort

Solution: b) Insertion sort

### Which sorting algorithm is used by the C++ STL sort() function?

- a) Quick sort
- b) Merge sort
- c) Heap sort
- d) Bubble sort

Solution: a) Quick sort

#### Which sorting algorithm is often used for sorting linked lists?

- a) Quick sort
- b) Merge sort
- c) Bubble sort
- d) Selection sort

Solution: b) Merge sort

## Which of the following sorting algorithms has a worst-case time complexity of O(n log n)?

- a) Quick sort
- b) Bubble sort
- c) Insertion sort
- d) Selection sort

Solution: a) Quick sort

## Which sorting algorithm works by repeatedly finding the minimum element from the unsorted part of the array and putting it at the beginning?

- a) Merge sort
- b) Quick sort
- c) Selection sort

d) Bubble sort

Solution: c) Selection sort

#### Which of the following is a disadvantage of using quick sort?

- a) Worst-case time complexity is O(n^2)
- b) It is not a comparison-based sorting algorithm
- c) It requires extra space for the temporary array
- d) It is not an in-place sorting algorithm

Solution: a) Worst-case time complexity is O(n^2)

#### Which sorting algorithm can be used for sorting strings in lexicographic order?

- a) Bubble sort
- b) Quick sort
- c) Insertion sort
- d) Radix sort

Solution: d) Radix sort

#### Which sorting algorithm is based on the divide-and-conquer strategy?

- a) Bubble sort
- b) Selection sort
- c) Merge sort
- d) Quick sort

Solution: c) Merge sort

### **Lec 5 - Linear Time Sorting**

#### 1. Which of the following is a linear time sorting algorithm?

- a) Quick sort
- b) Merge sort
- c) Counting sort
- d) Selection sort

Answer: c) Counting sort

### Which of the following is not a linear time sorting algorithm?

- a) Bucket sort
- b) Radix sort
- c) Quick sort
- d) All of the above

Answer: c) Quick sort

### What is the worst-case time complexity of counting sort?

- a) O(n)
- b) O(n log n)
- c) O(n^2)
- d) It depends on the input

Answer: a) O(n)

### Which of the following sorting algorithms is not comparison-based?

- a) Bucket sort
- b) Radix sort
- c) Quick sort
- d) All of the above

Answer: d) All of the above

### Which of the following is an advantage of linear time sorting algorithms?

- a) They have a faster runtime than comparison-based sorting algorithms.
- b) They work for all types of data.
- c) They have a lower memory usage than comparison-based sorting algorithms.
- d) They are more accurate than comparison-based sorting algorithms.

Answer: a) They have a faster runtime than comparison-based sorting algorithms.

#### Which of the following sorting algorithms is based on dividing elements into buckets?

- a) Counting sort
- b) Radix sort
- c) Bucket sort
- d) Selection sort

Answer: c) Bucket sort

#### Which of the following sorting algorithms is based on comparing digits or characters?

- a) Counting sort
- b) Radix sort
- c) Bucket sort
- d) Selection sort

Answer: b) Radix sort

### Which of the following sorting algorithms requires additional memory for the buckets?

a) Counting sort

- b) Radix sort
- c) Bucket sort
- d) Selection sort

## Answer: c) Bucket sort

# Which of the following is an example of an input that counting sort cannot sort in linear time?

- a) An array of integers
- b) A string of characters
- c) A binary tree
- d) A linked list

Answer: b) A string of characters

## Which of the following is not a stable sorting algorithm?

- a) Counting sort
- b) Radix sort
- c) Bucket sort
- d) Selection sort

Answer: d) Selection sort

### Lec 6 - Dynamic Programming

#### 1. Which of the following is not a characteristic of Dynamic Programming?

- A. It is a top-down approach
- B. It is based on recursion
- C. It involves solving subproblems only once
- D. It is a brute force technique

Answer: D

### Which of the following is the main goal of Dynamic Programming?

- A. To solve complex optimization problems
- B. To solve linear equations
- C. To simplify algorithms
- D. To generate random numbers

Answer: A

# Which of the following is an example of a problem that can be solved using Dynamic Programming?

- A. Sorting an array of integers
- B. Finding the shortest path in a graph
- C. Calculating the factorial of a number
- D. Generating random strings

**Answer: B** 

## What is the time complexity of Dynamic Programming?

- A. O(n)
- B. O(n log n)
- C. O(n^2)
- D. It depends on the problem being solved

**Answer: D** 

### Which of the following is a common technique used in Dynamic Programming?

- A. Binary search
- B. Bubble sort
- C. Memoization
- D. Quick sort

**Answer: C** 

### Which of the following is a disadvantage of using Dynamic Programming?

- A. It can be time-consuming
- B. It requires advanced mathematical knowledge
- C. It can lead to memory overflow
- D. It cannot be used for optimization problems

Answer: C

#### What is the difference between memoization and tabulation in Dynamic Programming?

- A. Memoization involves storing solutions in a table, while tabulation involves solving subproblems recursively.
- B. Memoization involves solving subproblems recursively, while tabulation involves storing solutions in a table.

- C. Memoization and tabulation are the same thing.
- D. Memoization is only used for top-down approaches, while tabulation is only used for bottomup approaches.

**Answer: B** 

# Which of the following is an example of a problem that can be solved using both recursive and iterative approaches?

- A. Finding the nth Fibonacci number
- B. Calculating the sum of an array of integers
- C. Sorting an array of integers
- D. Generating random numbers

**Answer: A** 

## Which of the following is not a step involved in solving a problem using Dynamic Programming?

- A. Breaking down the problem into smaller subproblems
- B. Storing the solutions of each subproblem in a table
- C. Solving the subproblems in a random order
- D. Combining the solutions of each subproblem to solve the original problem

**Answer: C** 

## Which of the following is a common optimization technique used in Dynamic Programming?

- A. Greedy algorithm
- B. Divide and conquer
- C. Backtracking
- D. Branch and bound

Answer: A

### Lec 7 - Greedy Algorithms

#### 1. Which of the following is a characteristic of Greedy algorithms?

- A) Always find the optimal solution
- B) Make locally optimal choices
- C) Require backtracking
- D) Can only be used for discrete problems

Answer: B) Make locally optimal choices

## Which of the following is an example of a problem that can be solved using a greedy algorithm?

- A) Traveling salesman problem
- B) Knapsack problem
- C) Graph coloring problem
- D) All of the above

Answer: B) Knapsack problem

#### Which of the following is a disadvantage of greedy algorithms?

- A) Always find the optimal solution
- B) May get stuck in local optima
- C) Are only useful for small problems
- D) Require exhaustive search

Answer: B) May get stuck in local optima

#### Which of the following is a common technique used to improve greedy algorithms?

- A) Dynamic programming
- B) Backtracking
- C) Randomization
- D) Exhaustive search

Answer: C) Randomization

#### Which of the following is an example of a greedy algorithm?

- A) Breadth-first search
- B) Depth-first search
- C) Dijkstra's algorithm
- D) Prim's algorithm

Answer: D) Prim's algorithm

## Which of the following is a necessary condition for a problem to be solved using a greedy algorithm?

- A) The problem must have optimal substructure
- B) The problem must be a minimization problem
- C) The problem must have only one solution
- D) The problem must be a continuous problem

Answer: A) The problem must have optimal substructure

## Which of the following is an example of a problem that cannot be solved using a greedy algorithm?

- A) Minimum spanning tree
- B) Shortest path problem

- C) Maximum flow problem
- D) Traveling salesman problem

#### Answer: D) Traveling salesman problem

#### Which of the following is a disadvantage of using a greedy algorithm?

- A) They are computationally expensive
- B) They always guarantee finding the optimal solution
- C) They require a lot of memory
- D) They may not always find the optimal solution

Answer: D) They may not always find the optimal solution

### Which of the following is a heuristic used in some greedy algorithms?

- A) Randomization
- B) Exhaustive search
- C) Divide and conquer
- D) Backtracking

Answer: A) Randomization

## Which of the following is an example of a problem that can be solved using a greedy algorithm with a proof of optimality?

- A) Huffman coding
- B) Fractional knapsack problem
- C) Job sequencing with deadlines
- D) All of the above

Answer: B) Fractional knapsack problem

### Lec 8 - Graphs

#### 1. What is a graph?

- A. A visual representation of data
- B. A mathematical structure used to model relationships
- C. A type of tree
- D. A type of function

#### Answer: B

#### What is a directed graph?

- A. A graph with loops
- B. A graph with weighted edges
- C. A graph with arrows on the edges
- D. A graph with multiple edges

#### **Answer: C**

#### What is an undirected graph?

- A. A graph with loops
- B. A graph with weighted edges
- C. A graph with arrows on the edges
- D. A graph with multiple edges

#### Answer: D

#### What is a weighted graph?

- A. A graph with loops
- B. A graph with arrows on the edges
- C. A graph with multiple edges
- D. A graph with values assigned to its edges

#### Answer: D

#### What is a cycle in a graph?

- A. A path from one vertex to another
- B. A connected component of a graph
- C. A sequence of vertices and edges that starts and ends at the same vertex
- D. A set of vertices that are not connected by any edge

#### **Answer: C**

#### What is a connected graph?

- A. A graph with no cycles
- B. A graph with all vertices connected by at least one edge
- C. A graph with multiple edges between vertices
- D. A graph with no loops

#### Answer: B

#### What is a tree?

- A. A type of graph with no cycles
- B. A type of graph with multiple edges
- C. A type of graph with loops
- D. A type of graph with weighted edges

#### Answer: A

#### What is a bipartite graph?

A. A graph with no cycles

- B. A graph with weighted edges
- C. A graph with two sets of vertices such that each edge connects a vertex from one set to a vertex in the other set
- D. A graph with multiple edges between vertices

#### **Answer: C**

#### What is a spanning tree?

- A. A tree that includes all vertices of a graph
- B. A tree with no cycles
- C. A tree with multiple edges between vertices
- D. A tree with weighted edges

#### Answer: A

#### What is the minimum spanning tree of a graph?

- A. The smallest tree that includes all vertices of the graph
- B. The tree with the minimum weight among all possible spanning trees of the graph
- C. The tree with the maximum weight among all possible spanning trees of the graph
- D. A tree that includes only a subset of the vertices of the graph

#### Answer: B

## **Lec 9 - Complexity Theory**

1.	What is the time complexity of binary search algorithm?  a. O(1)  b. O(log n)  c. O(n)  d. O(n^2)  Answer: b. O(log n)
	What is the space complexity of bubble sort algorithm?  a. O(1)  b. O(n)  c. O(n^2)  d. O(log n)  Answer: a. O(1)
	Which complexity class does the problem of factoring large integers belong to?  a. P  b. NP  c. NP-hard  d. NP-complete  Answer: d. NP-complete
	Which complexity class does the problem of finding the shortest path in a graph belong to?  a. P  b. NP  c. NP-hard  d. NP-complete  Answer: a. P
	What is the worst-case time complexity of the brute-force algorithm for the traveling salesman problem?  a. O(n!)  b. O(2^n) c. O(n^2) d. O(log n)  Answer: a. O(n!)
	Which of the following is not a complexity class?  a. PSPACE b. PTIME c. EXP d. NPSPACE Answer: d. NPSPACE
	What is the time complexity of the merge sort algorithm?  a. O(1)  b. O(n)  c. O(n log n)

d. O(n^2)

#### Answer: c. O(n log n)

### Which of the following is an example of a decision problem?

- a. Sorting a list of integers
- b. Finding the shortest path in a graph
- c. Determining whether a number is prime
- d. Factoring a large integer

Answer: c. Determining whether a number is prime

#### Which of the following complexity classes is believed to be strictly larger than P?

- a. NP
- b. PSPACE
- c. EXP
- d. NP-complete

Answer: c. EXP

### What is the time complexity of the naive algorithm for matrix multiplication?

- a. O(1)
- b. O(n)
- c. O(n^2)
- d. O(n^3)

Answer: d. O(n^3)