Question 1

You need the longest substring with all unique characters (e.g., "abcabcbb" -> 3).

A) Sorting characters first

B) BFS

C) Sliding window + Set/Map

D) Heap

Question 1

You need the longest substring with all unique characters (e.g., "abcabcbb" -> 3).

A) Sorting characters first

B) BFS

C) Sliding window + Set/Map

D) Heap

Solution

Correct option: C) Sliding window + Set/Map

Question 2

Given an unsorted array, find if any two numbers sum to target with best average performance.

A) Binary search each element

B) HashMap (store complements)

C) DFS

D) Two pointers without sorting

Question 2

Given an unsorted array, find if any two numbers sum to target with best average performance.

A) Binary search each element

B) HashMap (store complements)

C) DFS

D) Two pointers without sorting

Solution

Correct option: B) HashMap (store complements)

Question 3

Minimum length subarray with sum >= S, all numbers are positive.

A) HashMap prefix sums

B) Sliding window

C) Sorting + two pointers

D) DFS

Question 3

Minimum length subarray with sum >= S, all numbers are positive.

A) HashMap prefix sums

B) Sliding window

C) Sorting + two pointers

D) DFS

Solution

Correct option: B) Sliding window

Question 4

Reverse a singly linked list in O(n) time, O(1) space.

A) Queue

B) HashSet

C) Pointer rewiring (iterative)

D) Heap

Question 4

Reverse a singly linked list in O(n) time, O(1) space.

A) Queue

B) HashSet

C) Pointer rewiring (iterative)

D) Heap

Solution

Correct option: C) Pointer rewiring (iterative)

Question 5

Detect a cycle in a linked list with O(1) extra space.

A) Sort nodes by value

B) Use recursion depth check

C) Use HashSet of visited nodes

D) Slow/fast pointers (Floyd)

Question 5

Detect a cycle in a linked list with O(1) extra space.

A) Sort nodes by value

B) Use recursion depth check

C) Use HashSet of visited nodes

D) Slow/fast pointers (Floyd)

Solution

Correct option: D) Slow/fast pointers (Floyd)

Question 6

Top K frequent elements in an array (K is small compared to n).

A) Sort by value

B) HashMap + min-heap of size K

C) DFS

D) Two pointers

Question 6

Top K frequent elements in an array (K is small compared to n).

A) Sort by value

B) HashMap + min-heap of size K

C) DFS

D) Two pointers

Solution

Correct option: B) HashMap + min-heap of size K

Question 7

Validate if a binary tree is a BST (global correctness, not just parent-child).

A) Track range bounds (min/max) during DFS

B) Only compare parent and immediate children

C) Do level-order and check increasing

D) Heapify then compare

Question 7

Validate if a binary tree is a BST (global correctness, not just parent-child).

A) Track range bounds (min/max) during DFS

B) Only compare parent and immediate children

C) Do level-order and check increasing

D) Heapify then compare

Solution

Correct option: A) Track range bounds (min/max) during DFS

Question 8

You must print nodes level-by-level (tree) and also want to separate levels cleanly.

A) DFS recursion

B) BFS using a queue (track level size)

C) Sorting

D) HashMap only

Question 8

You must print nodes level-by-level (tree) and also want to separate levels cleanly.

A) DFS recursion

B) BFS using a queue (track level size)

C) Sorting

D) HashMap only

Solution

Correct option: B) BFS using a queue (track level size)

Question 9

Count connected components in an undirected graph given adjacency list.

A) Topological sort

B) Dijkstra

C) DFS/BFS from each unvisited node

D) Binary search

Question 9

Count connected components in an undirected graph given adjacency list.

A) Topological sort

B) Dijkstra

C) DFS/BFS from each unvisited node

D) Binary search

Solution

Correct option: C) DFS/BFS from each unvisited node

Question 13

Kth largest element in an unsorted array (single query, want better than full sort).

A) Merge sort

B) Min-heap of size K

C) Binary search

D) Two pointers

Question 13

Kth largest element in an unsorted array (single query, want better than full sort).

A) Merge sort

B) Min-heap of size K

C) Binary search

D) Two pointers

Solution

Correct option: B) Min-heap of size K

Question 14

You need to check if a path exists between two nodes in a graph (not weighted).

A) Sliding window

B) Hashing values

C) BFS/DFS

D) Sorting

Question 14

You need to check if a path exists between two nodes in a graph (not weighted).

A) Sliding window

B) Hashing values

C) BFS/DFS

D) Sorting

Solution

Correct option: C) BFS/DFS

Question 15

Find the middle of a linked list in one pass.

A) Fast/slow pointers

B) Binary search

C) Heap

D) Sorting

Question 15

Find the middle of a linked list in one pass.

A) Fast/slow pointers

B) Binary search

C) Heap

D) Sorting

Solution

Correct option: A) Fast/slow pointers

Question 16

Return the first non-repeating character in a string (preserve original order).

A) Sort then scan

B) Frequency map + second pass

C) Sliding window

D) BFS

Question 16

Return the first non-repeating character in a string (preserve original order).

A) Sort then scan

B) Frequency map + second pass

C) Sliding window

D) BFS

Solution

Correct option: B) Frequency map + second pass

Question 17

Merge two sorted arrays into one sorted array.

A) DFS

B) HashSet

C) Two pointers (merge process)

D) Heapify

Question 17

Merge two sorted arrays into one sorted array.

A) DFS

B) HashSet

C) Two pointers (merge process)

D) Heapify

Solution

Correct option: C) Two pointers (merge process)

Question 18

Given a graph, you want to traverse and record discovery order and also avoid revisiting nodes.

A) Stack only

B) BFS/DFS + visited[]

C) Two pointers

D) Sorting

Question 18

Given a graph, you want to traverse and record discovery order and also avoid revisiting nodes.

A) Stack only

B) BFS/DFS + visited[]

C) Two pointers

D) Sorting

Solution

Correct option: B) BFS/DFS + visited[]

Question 19

You need the maximum depth/height of a binary tree.

A) BFS with queue

B) HashMap

C) Two pointers

D) Sorting

Question 19

You need the maximum depth/height of a binary tree.

A) BFS with queue

B) HashMap

C) Two pointers

D) Sorting

Solution

Correct option: A) BFS with queue

Question 20

You need the diameter of a binary tree (longest path between any two nodes).

A) Two pointers

B) Use DFS to compute heights and update a global max

C) Kahn’s algorithm

D) Binary search

Question 20

You need the diameter of a binary tree (longest path between any two nodes).

A) Two pointers

B) Use DFS to compute heights and update a global max

C) Kahn’s algorithm

D) Binary search

Solution

Correct option: B) Use DFS to compute heights and update a global max