

DATA

STRUCTURES

IN

JS

JAVASCRIPT

The most comprehensive
compilation of frequently
asked DSA questions in
JavaScript.

**INTERVIEW
QUESTIONS**

By RoadsideCoder

TABLE OF CONTENTS

00 How to Follow this book?

01 Big O Notation

02 Arrays

03 Strings

04 Recursion & Backtracking

05 Stacks

06 Queues

07 Searching Algorithms

08 Sorting Algorithms

09 Linked List

10 Trees

11 Dynamic Programming

12 Graph



HOW TO FOLLOW THIS E-BOOK?

- Pick a question
- Click on  WATCH THIS VIDEO ON ARRAYS - [HTTPS://YOUTUBE.WATCH?V=ICADHMEHMZO](https://youtu.be/watch?v=icadhmehmzo) to understand that concept. (If the video has not been released yet for a topic, it will be soon)
- Try the question by clicking on this icon - 
- If you're unable to solve a question, go to ChatGPT & type this with link to that question -

Explain step by step in JS (provide me multiple solutions) -

<https://leetcode.com/problems/merge-sorted-array/>



- This will provide you with multiple solution for same problem including brute force and more optimised ones.

This is the perfect resource for anyone looking to crack their **Frontend DSA Rounds!**

Whether you're a **beginner** or an **experienced** programmer, this e-book is a **must-read** to crack your **DSA interviews**.

It is recommended to follow my Playlist on **DSA with JS on YouTube along the way to learn and practice in the best way!**

→ → **CLICK HERE TO WATCH** ← ←

**TIME & SPACE
COMPLEXITY**

JS

BIG

NOTATION



TIME COMPLEXITY

Time complexity refers to the amount of time an algorithm takes to run as a function of the input size. It gives an idea of how the algorithm's performance scales with increasing input.

Let's take a simple example of finding the sum of an array of numbers:

```
function sumArray(arr) {  
    let sum = 0;  
    for (let i = 0; i < arr.length; i++) {  
        sum += arr[i];  
    }  
    return sum;  
}
```

In this case, the time complexity of the `sumArray` function is $O(n)$, where ' n ' is the number of elements in the input array. This is because the loop iterates through each element in the array exactly once.

SPACE COMPLEXITY

Space complexity refers to the amount of memory an algorithm uses as a function of the input size. It indicates how much memory is required by the algorithm to execute.

Let's modify our previous example to demonstrate space complexity:

```
function createArray(size) {  
    let arr = [];  
    for (let i = 0; i < size; i++) {  
        arr.push(i);  
    }  
    return arr;  
}
```

In this case, the space complexity of the `createArray` function is $O(n)$, where ' n ' is the input size. This is because the size of the array created is directly proportional to the input size.



WATCH THIS VIDEO FOR INDEPTH EXPLANATION ON BIG O NOTATION -
[HTTPS://WWW.YOUTUBE.COM/WATCH?V=LAEXPVI1VRE](https://www.youtube.com/watch?v=LAEXPVI1VRE)



ARRAYS



WATCH THIS VIDEO ON ARRAYS -
[HTTPS://YOUTU.BE/WATCH?V=ICADHMEHMZO](https://youtu.be/watch?v=icadhmehmzo)

QUESTIONS	DIFFICULTY	LINK
Rotate Array by K	EASY	G
Second Largest Number	EASY	G
Remove Duplicates from Sorted Array	EASY	G
Maximum Sum Subarray	EASY	G
Two Sum	EASY	G
Container With Most Water	MEDIUM	G
Best Time to Buy and Sell Stock	EASY	G
Product of Array Except Self	MEDIUM	G
Merge Intervals	MEDIUM	G
Combination Sum	MEDIUM	G
Missing Number	EASY	G
Find All Numbers Disappeared in an Array	EASY	G

“



”

STRINGS



WATCH THIS VIDEO ON STRINGS -
[HTTPS://YOUTU.BE/WATCH?V=EREFTICNPRC](https://youtu.be/watch?v=EREFTICNPRC)

QUESTIONS	DIFFICULTY	LINK
Valid Palindrome	EASY	G
Reverse String	EASY	G
Valid Anagram	EASY	G
Hamming Distance	EASY	G
Longest Palindromic Substring	MEDIUM	G
Group Anagrams	MEDIUM	G
Reverse String ii	EASY	G
Longest Common Prefix	EASY	G
String to Integer (atoi)	MEDIUM	G
Palindrome Permutation	MEDIUM	G
Longest Substring Without Repeating Characters	MEDIUM	G

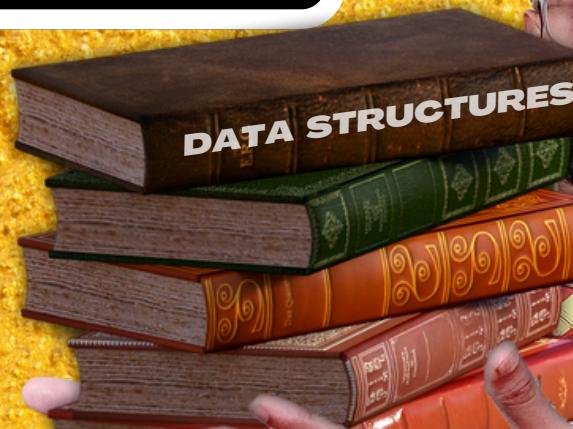
RECURSION AND BACKTRACKING

Always has been



QUESTIONS	DIFFICULTY	LINK
Fibonacci	EASY	G
Factorial Trailing Zeroses	MEDIUM	G
Create an array with range of numbers	EASY	G
Solve palindrome recursively	EASY	G
Reverse a String	EASY	G
Subsets	MEDIUM	G
Permutations	MEDIUM	G
Generate Parentheses	MEDIUM	G
Next Permutation	MEDIUM	G
Letter Combinations of a Phone Number	MEDIUM	G
Restore IP Addresses	MEDIUM	G
N-Queens	HARD	G
Sudoku Solver	HARD	G

STACKS



WATCH THIS VIDEO ON STACKS -

[HTTPS://YOUTU.BE/WATCH?V=UDHGN8UNO6W](https://youtu.be/watch?v=UDHGN8UNo6w)

QUESTIONS	DIFFICULTY	LINK
Reverse Words in a String	MEDIUM	G
Valid Parentheses	EASY	G
Min Stack	MEDIUM	G
Backspace String Compare	EASY	G
Basic Calculator	HARD	G
Decode String	MEDIUM	G
Evaluate Reverse Polish Notation	MEDIUM	G
Largest Rectangle in Histogram	HARD	G
Longest Valid Parentheses	HARD	G
Trapping Rain Water	HARD	G

QUEUE



WATCH THIS VIDEO ON QUEUE -

[HTTPS://YOUTU.BE/WATCH?V=SKTDEPXXRDW](https://youtu.be/watch?v=sktdepXXrdw)

QUESTIONS	DIFFICULTY	LINK
Implement Stack using Queues	EASY	🔗
Implement Queue using Stacks	EASY	🔗
Design Circular Queue	MEDIUM	🔗
Sliding Window Maximum	HARD	🔗
Design Hit Counter	MEDIUM	🔗

SEARCHING ALGORITHMS



WATCH THIS VIDEO ON SEARCH ALGOS -
[HTTPS: / YOUTU.BE / WATCH?V=-T8KE4KONFS](https://youtu.be/watch?v=-T8KE4konfs)

QUESTIONS	DIFFICULTY	LINK
Linear Search	EASY	🔗
Binary Search	EASY	🔗
Kth Missing Positive Number	EASY	🔗
Search in Rotated Sorted Array	EASY	🔗
Maximum Count of Positive Integer and Negative Integer	EASY	🔗
First Bad Version	EASY	🔗
Find Peak Element	MEDIUM	🔗
Find First and Last Position of Element in Sorted Array	MEDIUM	🔗
Search a 2D Matrix	MEDIUM	🔗
Count of Smaller Numbers After Self	HARD	🔗

BUBBLE SORT

INSERTION SORT

QUICK SORT

SELECTION SORT

MERGE SORT

SORTING ALGORITHMS



WATCH THIS VIDEO ON SORTING ALGOS -
[HTTPS: / YOUTU.BE / WATCH?V=NEKQXUYLQ2M](https://youtu.be/watch?v=NEKQXUYLQ2M)

QUESTIONS	DIFFICULTY	LINK
Sort an Array (Use Selection Sort)	MEDIUM	SOLVED
Sort Colors (Use Bubble Sort)	MEDIUM	SOLVED
Insertion Sort List (Use Insertion Sort)	MEDIUM	SOLVED
Merge Intervals (Use Merge sort)	MEDIUM	SOLVED
Sort an Array (Use Quick sort)	MEDIUM	SOLVED
Kth Largest Element in an Array	MEDIUM	SOLVED
Count of Smaller Numbers After Self	HARD	SOLVED
Find Minimum in Rotated Sorted Array	MEDIUM	SOLVED
Median of Two Sorted Arrays	HARD	SOLVED
Maximum Gap (Use Radix sort)	HARD	SOLVED

LINKED LIST



WATCH THIS VIDEO ON LINKED LIST -
[HTTPS://YOUTU.BE/WATCH?V=TJETC1F3EBY](https://youtu.be/watch?v=TJETC1F3EBY)

QUESTIONS	DIFFICULTY	LINK
Palindrome Linked List	EASY	🔗
Reverse Linked List	EASY	🔗
Remove Nth Node From End of List	MEDIUM	🔗
Add Two Numbers	MEDIUM	🔗
Flatten a Multilevel Doubly Linked List	MEDIUM	🔗
Delete Node in a Linked List	MEDIUM	🔗
Intersection of Two Linked Lists	EASY	🔗
LRU Cache	MEDIUM	🔗
Merge Two Sorted Lists	EASY	🔗
Linked List Cycle	EASY	🔗

TREES

WATCH THIS VIDEO ON TREES -
COMING SOOOOON

QUESTIONS	DIFFICULTY	LINK
Binary Tree Inorder Traversal	EASY	🔗
Binary Tree Preorder Traversal	EASY	🔗
Binary Tree Postorder Traversal	EASY	🔗
Path Sum	EASY	🔗
Symmetric Tree	EASY	🔗
Binary Tree Right Side View	MEDIUM	🔗
Maximum Width of Binary Tree	MEDIUM	🔗
Binary Tree Level Order Traversal	MEDIUM	🔗
Maximum Depth of Binary Tree	EASY	🔗
Invert Binary Tree	EASY	🔗
Validate Binary Search Tree	MEDIUM	🔗
Lowest Common Ancestor of a Binary Tree	MEDIUM	🔗

DYNAMIC PROGRAMMING



**WATCH THIS VIDEO ON DP -
COMING SOOOOON**

QUESTIONS	DIFFICULTY	LINK
Edit Distance	MEDIUM	🔗
Climbing Stairs	EASY	🔗
Best Time to Buy and Sell Stock	EASY	🔗
Coin Change	MEDIUM	🔗
Maximum Subarray	MEDIUM	🔗
Unique Paths	MEDIUM	🔗
Longest Increasing Subsequence	MEDIUM	🔗
Longest Palindromic Subsequence	MEDIUM	🔗
House Robber	MEDIUM	🔗
Word Break	MEDIUM	🔗

GRAPH



WATCH THIS VIDEO ON GRAPHS -
[COMING SOOOOON](#)

QUESTIONS	DIFFICULTY	LINK
Minimum Time to Collect All Apples in a Tree (DFS)	MEDIUM	🔗
Path with Maximum Probability (Dijkstra Algorithm)	MEDIUM	🔗
Shortest Path Visiting All Nodes (BFS)	HARD	🔗
Flood Fill	EASY	🔗
Clone Graph	MEDIUM	🔗
Number of Islands	MEDIUM	🔗
Course Schedule	MEDIUM	🔗
Network Delay Time	MEDIUM	🔗
Word Search	MEDIUM	🔗
Rotting Oranges	MEDIUM	🔗
Minimum Height Trees	MEDIUM	🔗
Word Ladder	HARD	🔗