

DSA Revision List: Pattern-wise High-ROI Questions

For JS developers targeting decent non-FAANG jobs. **LC** is the LeetCode number, ★ means must-do (if time is short, do only these), E = Easy, M = Medium, H = Hard.

Tier 1 ($\approx 60\%$ of questions)

1. Arrays + Hashing ($\approx 15\%$)

Question	LC	Level
★ Two Sum	1	E
★ Contains Duplicate	217	E
★ Valid Anagram	242	E
★ Group Anagrams	49	M
★ Top K Frequent Elements	347	M
★ Product of Array Except Self	238	M
Longest Consecutive Sequence	128	M
Majority Element	169	E
Missing Number	268	E
Maximum Subarray (Kadane)	53	M
Best Time to Buy and Sell Stock	121	E
Rotate Array	189	M

2. Two Pointers ($\approx 10\%$)

Question	LC	Level
★ Valid Palindrome	125	E

Question	LC	Level
★ Two Sum II (sorted array)	167	M
★ 3Sum	15	M
★ Container With Most Water	11	M
★ Move Zeroes	283	E
Remove Duplicates from Sorted Array	26	E
Reverse String	344	E
Sort Colors (Dutch flag)	75	M

3. Strings (≈10%)

Question	LC	Level
★ Longest Common Prefix	14	E
★ Reverse Words in a String	151	M
★ Longest Palindromic Substring	5	M
Is Subsequence	392	E
Isomorphic Strings	205	E
Valid Palindrome II	680	E
String Compression	443	M

4. Sliding Window (≈10%)

Question	LC	Level
★ Longest Substring Without Repeating Characters	3	M
★ Maximum Average Subarray I (fixed window)	643	E
★ Longest Repeating Character Replacement	424	M

Question	LC	Level
★ Permutation in String	567	M
Minimum Size Subarray Sum	209	M
Find All Anagrams in a String	438	M
Minimum Window Substring (stretch goal)	76	H

5. Binary Search + Sorting ($\approx 10\%$)

Question	LC	Level
★ Binary Search	704	E
★ Search in Rotated Sorted Array	33	M
★ Find Minimum in Rotated Sorted Array	153	M
★ Find First and Last Position	34	M
Search Insert Position	35	E
Sqrt(x)	69	E
Koko Eating Bananas	875	M
Merge Sorted Array	88	E

Tier 2 ($\approx 25\%$ of questions)

6. Stack ($\approx 6\%$)

Question	LC	Level
★ Valid Parentheses	20	E
★ Min Stack	155	M
★ Daily Temperatures	739	M
★ Next Greater Element I	496	E

Question	LC	Level
Evaluate Reverse Polish Notation	150	M
Implement Queue using Stacks	232	E
Remove All Adjacent Duplicates	1047	E

7. Linked List ($\approx 7\%$)

Question	LC	Level
★ Reverse Linked List	206	E
★ Linked List Cycle	141	E
★ Middle of the Linked List	876	E
★ Merge Two Sorted Lists	21	E
★ Remove Nth Node From End	19	M
Palindrome Linked List	234	E
Intersection of Two Linked Lists	160	E
Reorder List	143	M

8. Recursion basics ($\approx 4\%$)

Question	LC	Level
★ Fibonacci Number	509	E
★ Pow(x, n)	50	M
★ Flatten nested array (JS)	not on LC	-
Climbing Stairs	70	E
Reverse String (recursive)	344	E
Sum of digits, palindrome check (recursive)	not on LC	-

9. Trees ($\approx 8\%$)

Question	LC	Level
★ Maximum Depth of Binary Tree	104	E
★ Invert Binary Tree	226	E
★ Binary Tree Level Order Traversal	102	M
★ Validate BST	98	M
★ Lowest Common Ancestor of a BST	235	M
Same Tree	100	E
Symmetric Tree	101	E
Diameter of Binary Tree	543	E
Balanced Binary Tree	110	E
Path Sum	112	E
Binary Tree Right Side View	199	M
Kth Smallest Element in a BST	230	M
Inorder, Preorder, Postorder Traversal	94, 144, 145	E

Tier 3 ($\approx 10\%$ of questions)

10. Graphs: BFS/DFS only ($\approx 4\%$)

Question	LC	Level
★ Number of Islands	200	M
★ Flood Fill	733	E
★ Rotting Oranges	994	M
Max Area of Island	695	M
Clone Graph	133	M

11. Heap / Priority Queue ($\approx 2\%$)

Question	LC	Level
★ Kth Largest Element in an Array	215	M
★ Top K Frequent Elements (revisit)	347	M
Last Stone Weight	1046	E
K Closest Points to Origin	973	M

12. Backtracking, basic ($\approx 2\%$)

Question	LC	Level
★ Subsets	78	M
★ Permutations	46	M
Combination Sum	39	M

13. Prefix Sum + Intervals ($\approx 2\%$)

Question	LC	Level
★ Subarray Sum Equals K	560	M
★ Merge Intervals	56	M
Range Sum Query (Immutable)	303	E
Insert Interval	57	M
Meeting Rooms	252	E

JS-specific topics (don't skip)

These show up often in interviews for JS roles.

- Polyfills: `map`, `filter`, `reduce`, `bind`, `Promise.all`
- Debounce and throttle

- Deep clone, flatten object/array, currying
- Closures, `this`, event loop and output-prediction questions

Skip for now

- Dynamic Programming (only Climbing Stairs and Fibonacci as recursion practice)
- Segment Trees, Tries, Union-Find, Bit Manipulation
- Advanced graph algorithms (Dijkstra, Topological Sort)
- Hard-level problems

How to use this list

- **Total ★ questions:** about 75. Finish these before touching the others.
- **Per question:** try for 20-30 minutes, then read the solution, close it, and re-code it from scratch.
- **Revision sheet:** keep a table with columns *Question* / *Pattern* / *Date solved* / *Confident* (Y/N) / *Revise on*. Re-attempt anything marked N after 3 days and again after 10 days.
- Solve in JS, and say your approach and time complexity out loud before coding, since interviewers look for that.

Suggested time split

- **Weeks 1-3:** Tier 1 (Arrays, Hashing, Two Pointers, Strings, Sliding Window, Binary Search)
- **Weeks 4-6:** Stack, Linked List, Recursion, Trees
- **Weeks 7-8:** Graphs BFS/DFS, Heap, Backtracking basics, then revision