JavaScript Algorithms & Data Structures - Interview Cheat Sheet

Timestamp	Topic	Explanation	Use Case	Time Complexity	Interview Tip
0:05:35	Loops in JS	For, while, for...of, for...in loops for iteration	Iterating over arrays, objects	O(n)	Be ready to compare loop types and their use cases
0:06:10	Sum of Natural Numbers	Add numbers from 1 to n	Math problems, cumulative totals	O(1) with formula, O(n) with loop	Know both loop and formula approach
0:09:12	Sum of Digits	Add all digits of a number	Digit-based calculations	O(log■■ n)	Practice using modulo and integer division
0:12:30	Number of Digits	Count digits in a number	Validation of inputs like phone numbers	O(log■■ n)	Be ready to solve without converting to string
0:15:37	Palindrome Numbers	Number reads same forwards and backwards	Validation checks	O(n)	Know how to reverse number/string and compare
0:20:23	Fibonacci Numbers	Sequence where each num is sum of previous two	Dynamic programming problems	O(2■) naive, O(n) with DP	Explain recursion vs iterative vs memoization
0:23:26	Missing Numbers	Find missing element in range	Data integrity checks	O(n)	Know XOR trick for optimal solution
0:27:09	Time Complexity	Measure algorithm efficiency	Performance tuning	-	Always justify Big-O in interviews
0:39:10	Remove TLE	Reduce algorithm execution time	Competitive programming	-	Identify bottlenecks and optimize loops
0:41:33	Arrays in JS	Ordered list of elements	Data storage and manipulation	O(1) access, O(n) search	Explain shallow vs deep copy
1:14:36	Strings in JS	Immutable sequences of characters	Text processing	O(n)	Know string methods and immutability
1:51:14	Valid Anagram	Check if two strings have same characters	Cryptography, puzzles	O(n)	Sort and compare vs frequency map
1:58:57	Longest Common Prefix	Find common starting substring	Autocomplete features	O(n·m)	Explain horizontal vs vertical scanning
2:02:18	Merge Strings Alternately	Alternate characters from two strings	Data merging	O(n)	Handle different length strings
2:06:21	Length of Last Word	Find length of last word in a sentence	Text formatting	O(n)	Trim spaces before counting
2:11:26	Recursion	Function calling itself	Divide & conquer problems	Depends on recursion depth	Know base case importance
2:44:16	Linear Search	Check each element until found	Small dataset search	O(n)	Explain why it's bad for large datasets
2:49:53	First Occurrence in String	Find first match index	Text parsing	O(n)	Know indexOf and manual loop approach
3:03:23	Binary Search (Iterative)	Search sorted array by halving	Fast search	O(log n)	Know conditions for sorted data
3:19:16	Binary Search (Recursion)	Recursive binary search	Divide & conquer search	O(log n)	Be clear on base and recursive cases
3:24:44	Floor & Ceil Value	Largest ≤ X, smallest ≥ X	Approximation problems	O(log n)	Explain use in range queries

Timestamp	Topic	Explanation	Use Case	Time Complexity	Interview Tip
3:36:44	Objects in JS	Key-value storage	Data mapping	O(1) avg	Explain prototype chain
4:25:57	JavaScript sort()	Sorts array in place	Ordering data	O(n log n)	Know default sort is lexicographic
4:34:18	Bubble Sort	Compare and swap adjacent elements	Educational purposes	$O(n^2)$	Explain why it's inefficient
4:45:44	Selection Sort	Select min and swap	Small datasets	$O(n^2)$	Compare to bubble sort
4:54:44	Insertion Sort	Insert element in sorted portion	Nearly sorted data	$O(n^2)$	Explain why good for small datasets
5:06:49	Merge Sort	Divide and merge sorted halves	Large dataset sorting	O(n log n)	Explain stability
5:40:15	Quick Sort	Partition around pivot	High-performance sort	O(n log n) avg, $O(n^2)$ worst	Explain pivot choice
5:52:10	Map in JS	Key-value store with any key type	Caching	O(1) avg	Compare with Object
6:00:00	Sort Characters by Frequency	Order characters by count	Text analysis	O(n log n)	Know frequency map approach
6:15:37	Objects vs Map	Differences in features/performance	Choosing right data structure	-	Performance trade-offs
6:19:46	WeakMap in JS	Map with weakly referenced keys	Memory-sensitive mapping	O(1) avg	No iteration support
6:27:56	Set in JS	Collection of unique values	Duplicate removal	O(1) avg	Know conversion from array
6:40:16	Intersection of Two Arrays	Find common elements	Data comparison	O(n + m)	Use sets for optimization
6:53:07	Linked List in JS	Nodes linked by pointers	Dynamic data structures	O(n) traversal	Compare with arrays
7:15:16	Middle of Linked List	Find center node	Splitting lists	O(n)	Use slow & fast pointer technique
7:20:32	Linked List Cycle	Detect loops in list	Memory leak detection	O(n)	Floyd’s cycle detection
7:24:40	Reverse Linked List	Reverse node order	Data processing	O(n)	Iterative vs recursive
7:29:37	Stack in JS	LIFO structure	Undo operations	O(1) push/pop	Array vs linked list stack
7:33:16	Stack using Linked List	Stack with linked list	Memory flexibility	O(1) push/pop	When array stack is not suitable
7:35:30	Valid Parentheses	Check bracket correctness	Syntax validation	O(n)	Use stack
7:41:41	Queue in JS	FIFO structure	Scheduling tasks	O(1) enqueue/dequeue	Shift in array is O(n)
7:48:46	Queue using Linked List	Queue with linked list	Memory flexibility	O(1) enqueue/dequeue	Better than array for large data
7:58:38	Queue using Stack	Queue from stacks	Algorithm practice	O(n) enqueue	Two stack trick

Timestamp	Topic	Explanation	Use Case	Time Complexity	Interview Tip
8:05:24	Circular Queue	Queue wraps around	Fixed-size buffers	O(1)	Explain modulo operation