

Complete Array Mastery - From Scratch to Expert

PHASE 1: FUNDAMENTALS (Week 1-2)

Topics to Master:

- Array basics and memory representation
- Time/Space complexity of array operations
- Basic traversal patterns
- Simple search and manipulation

Essential Problems:

1. **[1480] Running Sum of 1d Array** - Basic array traversal
2. **[1672] Richest Customer Wealth** - Nested array basics
3. **[1431] Kids With the Greatest Number of Candies** - Array comparison
4. **[1365] How Many Numbers Are Smaller Than the Current Number** - Nested loops
5. **[1295] Find Numbers with Even Number of Digits** - Number manipulation
6. **[977] Squares of a Sorted Array** - Basic two pointers intro

Goal: Comfortable with basic array operations and thinking in terms of indices.

PHASE 2: SEARCHING & SORTING (Week 3-4)

Topics to Master:

- Linear search variations
- Binary search on arrays
- Basic sorting algorithms understanding
- Search in modified arrays

Essential Problems:

1. **[704] Binary Search** - Classic binary search
2. **[35] Search Insert Position** - Modified binary search
3. **[34] Find First and Last Position of Element in Sorted Array** - Binary search variants
4. **[33] Search in Rotated Sorted Array** - Advanced binary search
5. **[153] Find Minimum in Rotated Sorted Array** - Peak finding

6. **[162] Find Peak Element** - Binary search applications

Goal: Master binary search and recognize when/how to apply it.

PHASE 3: TWO POINTERS MASTERY (Week 5-6)

Topics to Master:

- Left-right pointer technique
- Fast-slow pointer (Floyd's algorithm)
- Multiple pointers coordination
- Partitioning problems

Essential Problems:

1. **[167] Two Sum II - Input Array Is Sorted** - Basic two pointers
2. **[15] 3Sum** - Three pointers with duplicates
3. **[11] Container With Most Water** - Optimization with pointers
4. **[42] Trapping Rain Water** - Complex pointer logic
5. **[75] Sort Colors** - Dutch National Flag (3-way partition)
6. **[26] Remove Duplicates from Sorted Array** - In-place modification

Goal: Instantly recognize two-pointer opportunities and implement efficiently.

PHASE 4: SLIDING WINDOW TECHNIQUES (Week 7-8)

Topics to Master:

- Fixed window size problems
- Variable window size problems
- Window with constraints
- Multiple condition tracking

Essential Problems:

1. **[643] Maximum Average Subarray I** - Fixed window
2. **[209] Minimum Size Subarray Sum** - Variable window
3. **[3] Longest Substring Without Repeating Characters** - Hash set tracking
4. **[424] Longest Repeating Character Replacement** - Window with replacements

5. **[76] Minimum Window Substring** - Complex condition tracking
6. **[567] Permutation in String** - Anagram in window

Goal: Recognize sliding window patterns and optimize brute force solutions.

PHASE 5: PREFIX SUM & RANGE QUERIES (Week 9-10)

Topics to Master:

- Basic prefix sum technique
- Hash map + prefix sum optimization
- 2D prefix sums
- Range query optimization

Essential Problems:

1. **[303] Range Sum Query - Immutable** - Basic prefix sum
2. **[560] Subarray Sum Equals K** - Hash map optimization
3. **[523] Continuous Subarray Sum** - Modular arithmetic
4. **[525] Contiguous Array** - Binary array prefix sum
5. **[930] Binary Subarrays With Sum** - Multiple valid subarrays
6. **[1442] Count Triplets That Can Form Two Arrays of Equal XOR** - XOR prefix

Goal: Optimize $O(n^2)$ range problems to $O(n)$ using prefix techniques.

PHASE 6: SUBARRAY PROBLEMS & KADANE'S (Week 11-12)

Topics to Master:

- Maximum/minimum subarray problems
- Kadane's algorithm and variations
- Circular array problems
- Multiple constraint subarray problems

Essential Problems:

1. **[53] Maximum Subarray** - Classic Kadane's
2. **[152] Maximum Product Subarray** - Handling negatives
3. **[918] Maximum Sum Circular Subarray** - Circular arrays

4. **[121] Best Time to Buy and Sell Stock** - State-based Kadane's
5. **[134] Gas Station** - Circular subarray application
6. **[862] Shortest Subarray with Sum at Least K** - Deque optimization

Goal: Master all variations of subarray optimization problems.

PHASE 7: HASH TABLE + ARRAY COMBINATIONS (Week 13-14)

Topics to Master:

- Frequency counting with arrays
- Multiple array coordination
- Anagram and permutation problems
- Duplicate detection strategies

Essential Problems:

1. **[1] Two Sum** - Hash map for pair finding
2. **[49] Group Anagrams** - String arrays + hashing
3. **[128] Longest Consecutive Sequence** - Set for sequence building
4. **[442] Find All Duplicates in an Array** - Array as hash table
5. **[41] First Missing Positive** - Array index as hash
6. **[347] Top K Frequent Elements** - Frequency + heap alternative

Goal: Efficiently combine hashing with array operations.

PHASE 8: ADVANCED PATTERNS (Week 15-16)

Topics to Master:

- Stack-based array problems
- Monotonic stack/deque
- Next greater/smaller element patterns
- Advanced optimization techniques

Essential Problems:

1. **[739] Daily Temperatures** - Monotonic stack
2. **[84] Largest Rectangle in Histogram** - Stack optimization

3. **[85] Maximal Rectangle** - 2D histogram
4. **[239] Sliding Window Maximum** - Deque for optimization
5. **[456] 132 Pattern** - Stack with multiple conditions
6. **[503] Next Greater Element II** - Circular array + stack

Goal: Handle complex optimization problems requiring auxiliary data structures.

PHASE 9: MATRIX & 2D ARRAYS (Week 17-18)

Topics to Master:

- 2D array traversal patterns
- Matrix rotation and transformation
- Search in 2D matrices
- Dynamic programming on matrices

Essential Problems:

1. **[54] Spiral Matrix** - Complex traversal pattern
2. **[48] Rotate Image** - In-place matrix rotation
3. **[73] Set Matrix Zeroes** - Matrix modification
4. **[240] Search a 2D Matrix II** - Efficient 2D search
5. **[79] Word Search** - Backtracking on matrix
6. **[200] Number of Islands** - Matrix DFS/BFS

Goal: Handle multi-dimensional array problems with confidence.

PHASE 10: INTERVIEW-LEVEL MASTERY (Week 19-20)

Topics to Master:

- Problem pattern recognition
- Multiple pattern combinations
- Optimization trade-offs
- Edge case handling mastery

Challenge Problems:




1. **[4] Median of Two Sorted Arrays** - Advanced binary search

2. **[10] Regular Expression Matching** - DP on strings/arrays
3. **[23] Merge k Sorted Lists** - Multiple array merging
4. **[32] Longest Valid Parentheses** - Stack + DP combination
5. **[135] Candy** - Multiple constraint optimization
6. **[315] Count of Smaller Numbers After Self** - Advanced data structures




Goal: Solve any array problem within 20-30 minutes in interview setting.

SUCCESS METRICS BY PHASE




Phase 1-2 (Beginner):

-  Solve basic array problems in 10-15 minutes
-  Understand time/space complexity
-  Handle edge cases (empty arrays, single elements)

Phase 3-6 (Intermediate):

-  Recognize patterns within 2-3 minutes
-  Optimize brute force solutions automatically
-  Explain approach before coding

Phase 7-10 (Advanced):

-  Solve complex problems in 20-30 minutes
 -  Handle follow-up questions and variations
 -  Combine multiple patterns seamlessly
-

PRACTICE STRATEGY

Daily Routine:

- **Week 1-10:** 1-2 problems per day from current phase
- **Week 11-15:** 2-3 problems per day, mix of review and new
- **Week 16-20:** 3-4 problems per day, focus on speed and accuracy

Weekly Review:

- Solve 5 random problems from previous phases

- Time yourself on each problem
- Practice explaining solutions out loud

Mock Interview Preparation:

- After Phase 6: Start weekly mock interviews
- Focus on communication and problem-solving process
- Practice with time pressure

Final Preparation:

- Last 2 weeks: Daily practice with mixed difficulty
- Focus on most commonly asked patterns
- Review your solution templates

By following this 20-week plan, you'll have the pattern recognition, implementation speed, and problem-solving confidence to tackle any array question in technical interviews.