

# LeetCode 200: Interview Mastery Path

By Engineering Digest

---

## 1. Arrays & Strings

### Easy

- #1 – [Two Sum](#)
- #121 – [Best Time to Buy and Sell Stock](#)
- #217 – [Contains Duplicate](#)
- #242 – [Valid Anagram](#)
- #125 – [Valid Palindrome](#)
- #344 – [Reverse String](#)
- #14 – [Longest Common Prefix](#)
- #53 – [Maximum Subarray](#)

### Medium

- #8 – [String to Integer \(atoi\)](#)
- #3 – [Longest Substring Without Repeating Characters](#)
- #49 – [Group Anagrams](#)
- #11 – [Container With Most Water](#)
- #15 – [3Sum](#)
- #33 – [Search in Rotated Sorted Array](#)
- #153 – [Find Minimum in Rotated Sorted Array](#)
- #238 – [Product of Array Except Self](#)
- #739 – [Daily Temperatures](#)
- #5 – [Longest Palindromic Substring](#)
- #56 – [Merge Intervals](#)

### Hard

- #76 – [Minimum Window Substring](#)
  - #57 – [Insert Interval](#)
  - #42 – [Trapping Rain Water](#)
  - #224 – [Basic Calculator](#)
-

## 2. Binary Search & Sorting

### Easy

- #278 – [First Bad Version](#)
- #35 – [Search Insert Position](#)

### Medium

- #34 – [Find First and Last Position of Element in Sorted Array](#)
- #162 – [Find Peak Element](#)

### Hard

- #4 – [Median of Two Sorted Arrays](#)
- 

## 3. Two Pointers & Sliding Window

### Easy

- #26 – [Remove Duplicates from Sorted Array](#)
- #283 – [Move Zeroes](#)
- #977 – [Squared Sorted Array](#)

### Medium

- #16 – [3Sum Closest](#)
- #18 – [4Sum](#)
- #438 – [Find All Anagrams in a String](#)
- #567 – [Permutation in String](#)
- #424 – [Longest Repeating Character Replacement](#)
- #209 – [Minimum Size Subarray Sum](#)
- #1004 – [Max Consecutive Ones III](#)
- #75 – [Sort Colors](#)

### Hard

- #239 – [Sliding Window Maximum](#)
  - #992 – [Subarrays with K Different Integers](#)
- 

## 4. Stacks & Queues

### Easy

- #20 – [Valid Parentheses](#)
- #155 – [Min Stack](#)
- #232 – [Implement Queue using Stacks](#)

#### Medium

- #150 – [Evaluate Reverse Polish Notation](#)
- 

## 5. Linked Lists

#### Easy

- #21 – [Merge Two Sorted Lists](#)
- #141 – [Linked List Cycle](#)
- #83 – [Remove Duplicates from Sorted List](#)
- #234 – [Palindrome Linked List](#)
- #876 – [Middle of the Linked List](#)
- #160 – [Intersection of Two Linked Lists](#)
- #206 – [Reverse Linked List](#)

#### Medium

- #2 – [Add Two Numbers](#)
- #19 – [Remove Nth Node From End of List](#)
- #24 – [Swap Nodes in Pairs](#)
- #61 – [Rotate List](#)
- #82 – [Remove Duplicates from Sorted List II](#)
- #86 – [Partition List](#)
- #138 – [Copy List with Random Pointer](#)
- #142 – [Linked List Cycle II](#)
- #143 – [Reorder List](#)
- #148 – [Sort List](#)

#### Hard

- #25 – [Reverse Nodes in k-Group](#)
  - #23 – [Merge k Sorted Lists](#)
- 

## 6. Trees (Binary Trees & BSTs)

### Easy

- #100 – [Same Tree](#)
- #101 – [Symmetric Tree](#)
- #104 – [Maximum Depth of Binary Tree](#)
- #112 – [Path Sum](#)
- #226 – [Invert Binary Tree](#)
- #572 – [Subtree of Another Tree](#)

### Medium

- #98 – [Validate Binary Search Tree](#)
- #102 – [Binary Tree Level Order Traversal](#)
- #103 – [Binary Tree Zigzag Level Order Traversal](#)
- #105 – [Construct Binary Tree from Preorder and Inorder Traversal](#)
- #113 – [Path Sum II](#)
- #114 – [Flatten Binary Tree to Linked List](#)
- #129 – [Sum Root to Leaf Numbers](#)
- #199 – [Binary Tree Right Side View](#)
- #230 – [Kth Smallest Element in a BST](#)
- #235 – [Lowest Common Ancestor of a Binary Tree](#)
- #337 – [House Robber III](#)
- #437 – [Path Sum III](#)
- #538 – [Convert BST to Greater Tree](#)
- #863 – [All Nodes Distance K in Binary Tree](#)
- #987 – [Vertical Order Traversal of a Binary Tree](#)
- #662 – [Maximum Width of Binary Tree](#)
- #1145 – [Binary Tree Coloring Game](#)

### Hard

- #99 – [Recover Binary Search Tree](#)
  - #124 – [Binary Tree Maximum Path Sum](#)
  - #297 – [Serialize and Deserialize Binary Tree](#)
  - #968 – [Binary Tree Cameras](#)
- 

## 7. Graphs

### Easy

- #323 – [Number of Connected Components in an Undirected Graph](#)
- #997 – [Find the Town Judge](#)

### Medium

- #200 – [Number of Islands](#)
- #130 – [Surrounded Regions](#)

- #133 – [Clone Graph](#)
- #207 – [Course Schedule](#)
- #210 – [Course Schedule II](#)
- #261 – [Graph Valid Tree](#)
- #399 – [Evaluate Division](#)
- #417 – [Pacific Atlantic Water Flow](#)
- #743 – [Network Delay Time](#)
- #785 – [Is Graph Bipartite?](#)
- #841 – [Keys and Rooms](#)
- #886 – [Possible Bipartition](#)
- #994 – [Rotting Oranges](#)
- #787 – [Cheapest Flights Within K Stops](#)
- #684 – [Redundant Connection](#)
- #947 – [Most Stones Removed with Same Row or Column](#)

Hard

- #127 – [Word Ladder](#)
  - #269 – [Alien Dictionary](#)
  - #212 – [Word Search II](#)
  - #1192 – [Critical Connections in a Network](#)
  - #721 – [Accounts Merge](#)
- 

## 8. Heaps & Priority Queues

Medium

- #347 – [Top K Frequent Elements](#)
- #378 – [Kth Smallest Element in a Sorted Matrix](#)
- #973 – [K Closest Points to Origin](#)

Hard

- #295 – [Find Median from Data Stream](#)
- 

## 9. Greedy

Medium

- #134 – [Gas Station](#)
- #45 – [Jump Game II](#)
- #406 – [Queue Reconstruction by Height](#)
- #452 – [Minimum Number of Arrows to Burst Balloons](#)
- #621 – [Task Scheduler](#)

Hard

- #135 – [Candy](#)
- 

## 10. Dynamic Programming

Easy

- #70 – [Climbing Stairs](#)

Medium

- #91 – [Decode Ways](#)
- #62 – [Unique Paths](#)
- #55 – [Jump Game](#)
- #152 – [Maximum Product Subarray](#)
- #198 – [House Robber](#)
- #213 – [House Robber II](#)
- #300 – [Longest Increasing Subsequence](#)
- #139 – [Word Break](#)
- #377 – [Combination Sum IV](#)
- #322 – [Coin Change](#)
- #647 – [Palindromic Substrings](#)
- #516 – [Longest Palindromic Subsequence](#)
- #64 – [Minimum Path Sum](#)
- #740 – [Delete and Earn](#)
- #718 – [Maximum Length of Repeated Subarray](#)
- #309 – [Best Time to Buy and Sell Stock with Cooldown](#)
- #494 – [Target Sum](#)
- #416 – [Partition Equal Subset Sum](#)
- #279 – [Perfect Squares](#)
- #1143 – [Longest Common Subsequence](#)
- #1048 – [Longest String Chain](#)
- #877 – [Stone Game](#)
- #97 – [Interleaving String](#)

Hard

- #72 – [Edit Distance](#)
- #10 – [Regular Expression Matching](#)
- #44 – [Wildcard Matching](#)
- #312 – [Burst Balloons](#)
- #85 – [Maximal Rectangle](#)
- #115 – [Distinct Subsequences](#)
- #87 – [Scramble String](#)
- #730 – [Count Different Palindromic Subsequences](#)

- #1000 – [Minimum Cost to Merge Stones](#)
  - #920 – [Number of Music Playlists](#)
  - #689 – [Maximum Sum of 3 Non-Overlapping Subarrays](#)
- 

## 11. Backtracking

### Medium

- #17 – [Letter Combinations of a Phone Number](#)
- #39 – [Combination Sum](#)
- #40 – [Combination Sum II](#)
- #46 – [Permutations](#)
- #47 – [Permutations II](#)
- #78 – [Subsets](#)
- #90 – [Subsets II](#)
- #77 – [Combinations](#)
- #131 – [Palindrome Partitioning](#)
- #93 – [Restore IP Addresses](#)
- #526 – [Beautiful Arrangement](#)
- #95 – [Unique Binary Search Trees II](#)

### Hard

- #51 – [N-Queens](#)
  - #37 – [Sudoku Solver](#)
  - #301 – [Remove Invalid Parentheses](#)
  - #282 – [Expression Add Operators](#)
  - #140 – [Word Break II](#)
  - #79 – [Word Search](#)
- 

## 12. System Design

### Easy

- #706 – [Design HashMap](#)
- #1603 – [Design Parking System](#)
- #346 – [Moving Average from Data Stream](#)

### Medium

- #146 – [LRU Cache](#)
- #380 – [Insert Delete GetRandom O\(1\)](#)
- #208 – [Implement Trie \(Prefix Tree\)](#)
- #211 – [Design Add and Search Words Data Structure](#)

- #359 – [Logger Rate Limiter](#)
- #244 – [Shortest Word Distance II](#)
- #729 – [My Calendar I](#)
- #528 – [Random Pick with Weight](#)
- #1472 – [Design Browser History](#)
- #1396 – [Design Underground System](#)
- #622 – [Design Circular Queue](#)
- #641 – [Design Circular Deque](#)
- #362 – [Design Hit Counter](#)

## Hard

- #355 – [Design Twitter](#)
- #642 – [Design Search Autocomplete System](#)
- #895 – [Maximum Frequency Stack](#)
- #1032 – [Stream of Characters](#)
- #588 – [Design In-Memory File System](#)
- #353 – [Design Snake Game](#)
- #432 – [All O'one Data Structure](#)
- #1188 – [Design File System](#)