Table of Contents

[DSA Leetcode programs list 2](#_Toc185884379)

# DSA Leetcode programs list

|  |  |  |  |
| --- | --- | --- | --- |
| # | Topic | Question | Difficulty |
| 1 | **Array, Hash Table** | [https://leetcode.com/problems/two-sum](https://www.google.com/url?q=https://leetcode.com/problems/two-sum&sa=D&source=editors&ust=1707482066860672&usg=AOvVaw1vW66-UQ3W3bXfXlaAsaVr) | **Easy** |
| 2 | **Hash Table, String, Sliding Window** | [https://leetcode.com/problems/longest-substring-without-repeating-characters](https://www.google.com/url?q=https://leetcode.com/problems/longest-substring-without-repeating-characters&sa=D&source=editors&ust=1707482066860879&usg=AOvVaw1OS4jZK3oZQkDfowDffWVF) | **Medium** |
| 3 | **Array, Divide and Conquer, Dynamic Programming** | [https://leetcode.com/problems/maximum-subarray](https://www.google.com/url?q=https://leetcode.com/problems/maximum-subarray&sa=D&source=editors&ust=1707482066861065&usg=AOvVaw2zzHdx8VIJ59Gzg8ltBWLs) | **Easy** |
| 4 | **Array, Two Pointers, Dynamic Programming, Stack, Monotonic Stack** | [https://leetcode.com/problems/trapping-rain-water](https://www.google.com/url?q=https://leetcode.com/problems/trapping-rain-water&sa=D&source=editors&ust=1707482066861250&usg=AOvVaw3kmliJRN4wP2FjsU4G-ke-) | **Hard** |
| 5 | **Linked List, Math, Recursion** | [https://leetcode.com/problems/add-two-numbers](https://www.google.com/url?q=https://leetcode.com/problems/add-two-numbers&sa=D&source=editors&ust=1707482066861451&usg=AOvVaw2aK3OK6p0_JugMg3lus0qn) | **Medium** |
| 6 | **Array, Two Pointers, Sorting** | [https://leetcode.com/problems/3sum](https://www.google.com/url?q=https://leetcode.com/problems/3sum&sa=D&source=editors&ust=1707482066861637&usg=AOvVaw1MhkN4U16YjzRjlvnWjeKh) | **Medium** |
| 7 | **String, Dynamic Programming** | [https://leetcode.com/problems/longest-palindromic-substring](https://www.google.com/url?q=https://leetcode.com/problems/longest-palindromic-substring&sa=D&source=editors&ust=1707482066861825&usg=AOvVaw3CeAiTvdPcSxJ3bTtIqDoT) | **Medium** |
| 8 | **Array, Binary Search, Divide and Conquer** | [https://leetcode.com/problems/median-of-two-sorted-arrays](https://www.google.com/url?q=https://leetcode.com/problems/median-of-two-sorted-arrays&sa=D&source=editors&ust=1707482066862012&usg=AOvVaw0liyTSDhJWkA6qvK8Dz7Kd) | **Hard** |
| 9 | **Array, Two Pointers, Greedy** | [https://leetcode.com/problems/container-with-most-water](https://www.google.com/url?q=https://leetcode.com/problems/container-with-most-water&sa=D&source=editors&ust=1707482066862230&usg=AOvVaw2SV7xzD-CM1vsfwKDacBp0) | **Medium** |
| 10 | **Array, Dynamic Programming** | [https://leetcode.com/problems/best-time-to-buy-and-sell-stock](https://www.google.com/url?q=https://leetcode.com/problems/best-time-to-buy-and-sell-stock&sa=D&source=editors&ust=1707482066862412&usg=AOvVaw2EAW67Lw2o7ptQDa3bF2wN) | **Easy** |
| 11 | **Array, Binary Search** | [https://leetcode.com/problems/search-in-rotated-sorted-array](https://www.google.com/url?q=https://leetcode.com/problems/search-in-rotated-sorted-array&sa=D&source=editors&ust=1707482066862591&usg=AOvVaw1anvGTda0P9Iztdol9f5Ix) | **Medium** |
| 12 | **Array, Depth-First Search, Breadth-First Search, Union Find, Matrix** | [https://leetcode.com/problems/number-of-islands](https://www.google.com/url?q=https://leetcode.com/problems/number-of-islands&sa=D&source=editors&ust=1707482066862764&usg=AOvVaw2MsB-tAFVPOtz_cMtn_aZ_) | **Medium** |
| 13 | **Hash Table, Linked List, Design, Doubly-Linked List** | [https://leetcode.com/problems/lru-cache](https://www.google.com/url?q=https://leetcode.com/problems/lru-cache&sa=D&source=editors&ust=1707482066862941&usg=AOvVaw35MAOzFD8_4TktQOVtwHQd) | **Medium** |
| 14 | **Array, Sorting** | [https://leetcode.com/problems/merge-intervals](https://www.google.com/url?q=https://leetcode.com/problems/merge-intervals&sa=D&source=editors&ust=1707482066863073&usg=AOvVaw21lZU7BEblzjIJzi_l79oq) | **Medium** |
| 15 | **String, Dynamic Programming, Backtracking** | [https://leetcode.com/problems/generate-parentheses](https://www.google.com/url?q=https://leetcode.com/problems/generate-parentheses&sa=D&source=editors&ust=1707482066863191&usg=AOvVaw0u7lt2VdtMjxUpbl6umTxZ) | **Medium** |
| 16 | **Array, Two Pointers, Binary Search, Bit Manipulation** | [https://leetcode.com/problems/find-the-duplicate-number](https://www.google.com/url?q=https://leetcode.com/problems/find-the-duplicate-number&sa=D&source=editors&ust=1707482066863295&usg=AOvVaw1BNJd8Aj8u6XH3tVwet7ed) | **Medium** |
| 17 | **Array, Prefix Sum** | [https://leetcode.com/problems/product-of-array-except-self](https://www.google.com/url?q=https://leetcode.com/problems/product-of-array-except-self&sa=D&source=editors&ust=1707482066863458&usg=AOvVaw1z6TIWvhwNZlHRIB6Ipaje) | **Medium** |
| 18 | **String, Stack** | [https://leetcode.com/problems/valid-parentheses](https://www.google.com/url?q=https://leetcode.com/problems/valid-parentheses&sa=D&source=editors&ust=1707482066863678&usg=AOvVaw0zfA8XK19Zz8eWlTxhmROG) | **Easy** |
| 19 | **Array, Hash Table, Prefix Sum** | [https://leetcode.com/problems/subarray-sum-equals-k](https://www.google.com/url?q=https://leetcode.com/problems/subarray-sum-equals-k&sa=D&source=editors&ust=1707482066863893&usg=AOvVaw0bvc4i2Y6cd15P-0DhYU48) | **Medium** |
| 20 | **Array, Dynamic Programming** | [https://leetcode.com/problems/house-robber](https://www.google.com/url?q=https://leetcode.com/problems/house-robber&sa=D&source=editors&ust=1707482066864109&usg=AOvVaw2Qu6L0n_13p6W80SxfbSoX) | **Medium** |
| 21 | **Array, Binary Search, Dynamic Programming** | [https://leetcode.com/problems/longest-increasing-subsequence](https://www.google.com/url?q=https://leetcode.com/problems/longest-increasing-subsequence&sa=D&source=editors&ust=1707482066864330&usg=AOvVaw1v9codCvD-jT38Tmhd1sxq) | **Medium** |
| 22 | **Linked List, Recursion** | [https://leetcode.com/problems/reverse-linked-list](https://www.google.com/url?q=https://leetcode.com/problems/reverse-linked-list&sa=D&source=editors&ust=1707482066864524&usg=AOvVaw0gakLYoEZL4AHDYU9ziTXC) | **Easy** |
| 23 | **Array, Dynamic Programming** | [https://leetcode.com/problems/maximum-product-subarray](https://www.google.com/url?q=https://leetcode.com/problems/maximum-product-subarray&sa=D&source=editors&ust=1707482066864718&usg=AOvVaw3n02N4Z2EkgReF5s9wki0L) | **Medium** |
| 24 | **Linked List, Divide and Conquer, Heap (Priority Queue), Merge Sort** | [https://leetcode.com/problems/merge-k-sorted-lists](https://www.google.com/url?q=https://leetcode.com/problems/merge-k-sorted-lists&sa=D&source=editors&ust=1707482066864906&usg=AOvVaw0eaeNAENXV6kVTACLeIipG) | **Hard** |
| 25 | **Linked List, Recursion** | [https://leetcode.com/problems/merge-two-sorted-lists](https://www.google.com/url?q=https://leetcode.com/problems/merge-two-sorted-lists&sa=D&source=editors&ust=1707482066865106&usg=AOvVaw2BHa1rOm37ZjZ4dmv966vF) | **Easy** |
| 26 | **Math, Dynamic Programming, Memoization** | [https://leetcode.com/problems/climbing-stairs](https://www.google.com/url?q=https://leetcode.com/problems/climbing-stairs&sa=D&source=editors&ust=1707482066865327&usg=AOvVaw3LOjInU4dR7qtOaX6PjOAo) | **Easy** |
| 27 | **Array, Dynamic Programming, Breadth-First Search** | [https://leetcode.com/problems/coin-change](https://www.google.com/url?q=https://leetcode.com/problems/coin-change&sa=D&source=editors&ust=1707482066865551&usg=AOvVaw1ss9CcNjnkDMHnf3_6-cdA) | **Medium** |
| 28 | **Array, Dynamic Programming, Greedy** | [https://leetcode.com/problems/jump-game](https://www.google.com/url?q=https://leetcode.com/problems/jump-game&sa=D&source=editors&ust=1707482066865760&usg=AOvVaw2caCo4sbFyZ_Kh9cFTdBI6) | **Medium** |
| 29 | **Hash Table, String, Dynamic Programming, Trie, Memoization** | [https://leetcode.com/problems/word-break](https://www.google.com/url?q=https://leetcode.com/problems/word-break&sa=D&source=editors&ust=1707482066865942&usg=AOvVaw09y-RidvHzDI9h_NCXM44d) | **Medium** |
| 30 | **Hash Table, String, Sliding Window** | [https://leetcode.com/problems/minimum-window-substring](https://www.google.com/url?q=https://leetcode.com/problems/minimum-window-substring&sa=D&source=editors&ust=1707482066866114&usg=AOvVaw3zkWpze0jF6ghY3qkldByq) | **Hard** |
| 31 | **Array, Binary Search** | [https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array](https://www.google.com/url?q=https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array&sa=D&source=editors&ust=1707482066866295&usg=AOvVaw2_jmYp_miui_0URA3qGwDg) | **Medium** |
| 32 | **Array, Backtracking** | [https://leetcode.com/problems/permutations](https://www.google.com/url?q=https://leetcode.com/problems/permutations&sa=D&source=editors&ust=1707482066866459&usg=AOvVaw3b23EyJrWjmjpjnPWSZAWo) | **Medium** |
| 33 | **Array, Backtracking** | [https://leetcode.com/problems/combination-sum](https://www.google.com/url?q=https://leetcode.com/problems/combination-sum&sa=D&source=editors&ust=1707482066866602&usg=AOvVaw0XfsZ6W184SVita_L1XNG-) | **Medium** |
| 34 | **Hash Table, String, Backtracking** | [https://leetcode.com/problems/letter-combinations-of-a-phone-number](https://www.google.com/url?q=https://leetcode.com/problems/letter-combinations-of-a-phone-number&sa=D&source=editors&ust=1707482066866773&usg=AOvVaw3r004Am-IO8J5UA1I9eePD) | **Medium** |
| 35 | **Tree, Depth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/validate-binary-search-tree](https://www.google.com/url?q=https://leetcode.com/problems/validate-binary-search-tree&sa=D&source=editors&ust=1707482066866957&usg=AOvVaw2AmYszOsaW0eqh0sRwFxG5) | **Medium** |
| 36 | **Array, Queue, Sliding Window, Heap (Priority Queue), Monotonic Queue** | [https://leetcode.com/problems/sliding-window-maximum](https://www.google.com/url?q=https://leetcode.com/problems/sliding-window-maximum&sa=D&source=editors&ust=1707482066867115&usg=AOvVaw2C-1frDqToE4TpAE6SuOSU) | **Hard** |
| 37 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree&sa=D&source=editors&ust=1707482066867268&usg=AOvVaw2luSw6e1tJLM6vXup6lY7n) | **Medium** |
| 38 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/symmetric-tree](https://www.google.com/url?q=https://leetcode.com/problems/symmetric-tree&sa=D&source=editors&ust=1707482066867439&usg=AOvVaw3EOgZy8vERxf1xaACuN4E8) | **Easy** |
| 39 | **Array, Two Pointers** | [https://leetcode.com/problems/next-permutation](https://www.google.com/url?q=https://leetcode.com/problems/next-permutation&sa=D&source=editors&ust=1707482066867617&usg=AOvVaw1q77WeVmI8ltiCOH_4ETg8) | **Medium** |
| 40 | **Array, Two Pointers, Sorting** | [https://leetcode.com/problems/sort-colors](https://www.google.com/url?q=https://leetcode.com/problems/sort-colors&sa=D&source=editors&ust=1707482066867807&usg=AOvVaw3MHujTS1xbZ_H1C2NryJui) | **Medium** |
| 41 | **Linked List, Two Pointers** | [https://leetcode.com/problems/remove-nth-node-from-end-of-list](https://www.google.com/url?q=https://leetcode.com/problems/remove-nth-node-from-end-of-list&sa=D&source=editors&ust=1707482066867954&usg=AOvVaw2Z8CBnH_mdA6N6OhQ-rrIp) | **Medium** |
| 42 | **Array, Bit Manipulation** | [https://leetcode.com/problems/single-number](https://www.google.com/url?q=https://leetcode.com/problems/single-number&sa=D&source=editors&ust=1707482066868120&usg=AOvVaw2W4ZFomDpMhfZoP5ozQ53p) | **Easy** |
| 43 | **Array, Stack, Monotonic Stack** | [https://leetcode.com/problems/largest-rectangle-in-histogram](https://www.google.com/url?q=https://leetcode.com/problems/largest-rectangle-in-histogram&sa=D&source=editors&ust=1707482066868318&usg=AOvVaw39il798leiZC-0Dj3HcB8L) | **Hard** |
| 44 | **Array, Hash Table** | [https://leetcode.com/problems/first-missing-positive](https://www.google.com/url?q=https://leetcode.com/problems/first-missing-positive&sa=D&source=editors&ust=1707482066868478&usg=AOvVaw1cTurrPn6BvZcNlG6-2ta8) | **Hard** |
| 45 | **Depth-First Search, Breadth-First Search, Graph, Topological Sort** | [https://leetcode.com/problems/course-schedule](https://www.google.com/url?q=https://leetcode.com/problems/course-schedule&sa=D&source=editors&ust=1707482066868623&usg=AOvVaw0tRrf9VLd8H4Sayv1HelkC) | **Medium** |
| 46 | **Dynamic Programming, Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-maximum-path-sum](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-maximum-path-sum&sa=D&source=editors&ust=1707482066868765&usg=AOvVaw20-9IysoXI8hrgQUQcje_G) | **Hard** |
| 47 | **Array, Backtracking, Matrix** | [https://leetcode.com/problems/word-search](https://www.google.com/url?q=https://leetcode.com/problems/word-search&sa=D&source=editors&ust=1707482066868920&usg=AOvVaw2ZYgg7k5se85n16FPb7EDR) | **Medium** |
| 48 | **Array, Backtracking, Bit Manipulation** | [https://leetcode.com/problems/subsets](https://www.google.com/url?q=https://leetcode.com/problems/subsets&sa=D&source=editors&ust=1707482066869061&usg=AOvVaw2_JNB8NYFq3qZidHiIQr21) | **Medium** |
| 49 | **Array, Hash Table, Union Find** | [https://leetcode.com/problems/longest-consecutive-sequence](https://www.google.com/url?q=https://leetcode.com/problems/longest-consecutive-sequence&sa=D&source=editors&ust=1707482066869239&usg=AOvVaw0Pe08VwkfNPnklHKtdlvS4) | **Medium** |
| 50 | **Array, Divide and Conquer, Sorting, Heap (Priority Queue), Quickselect** | [https://leetcode.com/problems/kth-largest-element-in-an-array](https://www.google.com/url?q=https://leetcode.com/problems/kth-largest-element-in-an-array&sa=D&source=editors&ust=1707482066869400&usg=AOvVaw0WG6j-2EPjlBcebaF-pJp7) | **Medium** |
| 51 | **Hash Table, String, Sorting** | [https://leetcode.com/problems/group-anagrams](https://www.google.com/url?q=https://leetcode.com/problems/group-anagrams&sa=D&source=editors&ust=1707482066869578&usg=AOvVaw3WbD-AqjwS1v86fViIonUc) | **Medium** |
| 52 | **Array, Two Pointers** | [https://leetcode.com/problems/move-zeroes](https://www.google.com/url?q=https://leetcode.com/problems/move-zeroes&sa=D&source=editors&ust=1707482066869747&usg=AOvVaw2SPpemyboiGZccZ65tfH-P) | **Easy** |
| 53 | **Hash Table, Linked List, Two Pointers** | [https://leetcode.com/problems/intersection-of-two-linked-lists](https://www.google.com/url?q=https://leetcode.com/problems/intersection-of-two-linked-lists&sa=D&source=editors&ust=1707482066869919&usg=AOvVaw1R3MS180zvHowx5AUisIk4) | **Easy** |
| 54 | **Array, Hash Table, Divide and Conquer, Sorting, Counting** | [https://leetcode.com/problems/majority-element](https://www.google.com/url?q=https://leetcode.com/problems/majority-element&sa=D&source=editors&ust=1707482066870091&usg=AOvVaw0vlGs7ogaQfs2tmCivj-E3) | **Easy** |
| 55 | **Math, Dynamic Programming, Combinatorics** | [https://leetcode.com/problems/unique-paths](https://www.google.com/url?q=https://leetcode.com/problems/unique-paths&sa=D&source=editors&ust=1707482066870240&usg=AOvVaw397UMLQel-V75OMirg9nFi) | **Medium** |
| 56 | **Array, Math, Matrix** | [https://leetcode.com/problems/rotate-image](https://www.google.com/url?q=https://leetcode.com/problems/rotate-image&sa=D&source=editors&ust=1707482066870370&usg=AOvVaw2EVy2FE4csO66zrYb5RcO5) | **Medium** |
| 57 | **Linked List, Two Pointers, Stack, Recursion** | [https://leetcode.com/problems/palindrome-linked-list](https://www.google.com/url?q=https://leetcode.com/problems/palindrome-linked-list&sa=D&source=editors&ust=1707482066870494&usg=AOvVaw0XcedSBX5iZA20NRHUOmm1) | **Easy** |
| 58 | **String, Dynamic Programming** | [https://leetcode.com/problems/edit-distance](https://www.google.com/url?q=https://leetcode.com/problems/edit-distance&sa=D&source=editors&ust=1707482066870602&usg=AOvVaw2dxwgJCUD14B0WSjnPKD1G) | **Hard** |
| 59 | **String, Dynamic Programming, Recursion** | [https://leetcode.com/problems/regular-expression-matching](https://www.google.com/url?q=https://leetcode.com/problems/regular-expression-matching&sa=D&source=editors&ust=1707482066870729&usg=AOvVaw0pS4sg2K-CvlcJfK0owZkz) | **Hard** |
| 60 | **Array, Hash Table, Divide and Conquer, Tree, Binary Tree** | [https://leetcode.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal](https://www.google.com/url?q=https://leetcode.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal&sa=D&source=editors&ust=1707482066871093&usg=AOvVaw3ehVGVcihvvxmNgAIRpNWR) | **Medium** |
| 61 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/invert-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/invert-binary-tree&sa=D&source=editors&ust=1707482066871214&usg=AOvVaw35SqrIh8-tGRfpt4VkPWDT) | **Easy** |
| 62 | **Array, Math, Two Pointers** | [https://leetcode.com/problems/rotate-array](https://www.google.com/url?q=https://leetcode.com/problems/rotate-array&sa=D&source=editors&ust=1707482066871322&usg=AOvVaw3aJ0bPRnrimctQzbQjtVCQ) | **Medium** |
| 63 | **String, Stack, Recursion** | [https://leetcode.com/problems/decode-string](https://www.google.com/url?q=https://leetcode.com/problems/decode-string&sa=D&source=editors&ust=1707482066871492&usg=AOvVaw1Xk6Y9PdeLLShvhes5uToQ) | **Medium** |
| 64 | **Stack, Design** | [https://leetcode.com/problems/min-stack](https://www.google.com/url?q=https://leetcode.com/problems/min-stack&sa=D&source=editors&ust=1707482066871677&usg=AOvVaw3Q3p34IJE9iTfTzyUr1e4e) | **Easy** |
| 65 | **Array, Hash Table, Divide and Conquer, Sorting, Heap (Priority Queue), Bucket Sort, Counting, Quickselect** | [https://leetcode.com/problems/top-k-frequent-elements](https://www.google.com/url?q=https://leetcode.com/problems/top-k-frequent-elements&sa=D&source=editors&ust=1707482066871873&usg=AOvVaw22XY8PB4ldjbycM_Op8bux) | **Medium** |
| 66 | **Hash Table, Linked List** | [https://leetcode.com/problems/copy-list-with-random-pointer](https://www.google.com/url?q=https://leetcode.com/problems/copy-list-with-random-pointer&sa=D&source=editors&ust=1707482066872071&usg=AOvVaw1PhGLIE4tyavvwxnREhUds) | **Medium** |
| 67 | **String, Dynamic Programming, Stack** | [https://leetcode.com/problems/longest-valid-parentheses](https://www.google.com/url?q=https://leetcode.com/problems/longest-valid-parentheses&sa=D&source=editors&ust=1707482066872303&usg=AOvVaw3NtSfcK2ji1Wgo3rh_ODPX) | **Hard** |
| 68 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/path-sum-iii](https://www.google.com/url?q=https://leetcode.com/problems/path-sum-iii&sa=D&source=editors&ust=1707482066872473&usg=AOvVaw3CAXqXSAdOZKZFP6xtW3wY) | **Medium** |
| 69 | **Tree, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-level-order-traversal](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-level-order-traversal&sa=D&source=editors&ust=1707482066872612&usg=AOvVaw3SWhvakOjiCzVh8XbZ8WcL) | **Medium** |
| 70 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/diameter-of-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/diameter-of-binary-tree&sa=D&source=editors&ust=1707482066872762&usg=AOvVaw3ok1Au5TSQyrgsOgZ2djVZ) | **Easy** |
| 71 | **Array, Dynamic Programming, Greedy** | [https://leetcode.com/problems/jump-game-ii](https://www.google.com/url?q=https://leetcode.com/problems/jump-game-ii&sa=D&source=editors&ust=1707482066872893&usg=AOvVaw2VqvoOcLrRu5qYTElfGdtI) | **Medium** |
| 72 | **Array, Dynamic Programming** | [https://leetcode.com/problems/partition-equal-subset-sum](https://www.google.com/url?q=https://leetcode.com/problems/partition-equal-subset-sum&sa=D&source=editors&ust=1707482066873024&usg=AOvVaw2vZRt5S8DHvldw311Jnl4T) | **Medium** |
| 73 | **Stack, Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-inorder-traversal](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-inorder-traversal&sa=D&source=editors&ust=1707482066873132&usg=AOvVaw0D9igRLyjYAZDZhL48OP4D) | **Easy** |
| 74 | **Hash Table, String, Breadth-First Search** | [https://leetcode.com/problems/word-ladder](https://www.google.com/url?q=https://leetcode.com/problems/word-ladder&sa=D&source=editors&ust=1707482066873251&usg=AOvVaw1w6XNHYLP8f7RA3a8u0Q19) | **Hard** |
| 75 | **Math, Dynamic Programming, Tree, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/unique-binary-search-trees](https://www.google.com/url?q=https://leetcode.com/problems/unique-binary-search-trees&sa=D&source=editors&ust=1707482066873382&usg=AOvVaw04i-2zLNIeHaAbKymkm95R) | **Medium** |
| 76 | **Hash Table, Linked List, Two Pointers** | [https://leetcode.com/problems/linked-list-cycle](https://www.google.com/url?q=https://leetcode.com/problems/linked-list-cycle&sa=D&source=editors&ust=1707482066873533&usg=AOvVaw3Zug6dN_0pxsXUZIDtECK8) | **Easy** |
| 77 | **Array, Dynamic Programming, Matrix** | [https://leetcode.com/problems/maximal-square](https://www.google.com/url?q=https://leetcode.com/problems/maximal-square&sa=D&source=editors&ust=1707482066873707&usg=AOvVaw1mKil6CALShv4VGJ7vV0eD) | **Medium** |
| 78 | **String** | [https://leetcode.com/problems/longest-common-prefix](https://www.google.com/url?q=https://leetcode.com/problems/longest-common-prefix&sa=D&source=editors&ust=1707482066873900&usg=AOvVaw1UoB12V5D00FGXI-iZNgZu) | **Easy** |
| 79 | **Array, Dynamic Programming, Greedy** | [https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii](https://www.google.com/url?q=https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii&sa=D&source=editors&ust=1707482066874096&usg=AOvVaw0buE_OFdZLpJujg64IMFnO) | **Medium** |
| 80 | **Array, Dynamic Programming, Matrix** | [https://leetcode.com/problems/minimum-path-sum](https://www.google.com/url?q=https://leetcode.com/problems/minimum-path-sum&sa=D&source=editors&ust=1707482066874291&usg=AOvVaw1t1gX-CCOIYCK46-sgHaXS) | **Medium** |
| 81 | **Array, Binary Search, Divide and Conquer, Matrix** | [https://leetcode.com/problems/search-a-2d-matrix-ii](https://www.google.com/url?q=https://leetcode.com/problems/search-a-2d-matrix-ii&sa=D&source=editors&ust=1707482066874455&usg=AOvVaw3NEudWSjNoTJ_J5d_NhpLR) | **Medium** |
| 82 | **Math** | [https://leetcode.com/problems/reverse-integer](https://www.google.com/url?q=https://leetcode.com/problems/reverse-integer&sa=D&source=editors&ust=1707482066874625&usg=AOvVaw3VL7PwE7tBbMHITUfEoOjr) | **Medium** |
| 83 | **Array, Binary Search** | [https://leetcode.com/problems/search-insert-position](https://www.google.com/url?q=https://leetcode.com/problems/search-insert-position&sa=D&source=editors&ust=1707482066874809&usg=AOvVaw2WaeIbxve09fbfbqTY7ErD) | **Easy** |
| 84 | **Linked List, Stack, Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/flatten-binary-tree-to-linked-list](https://www.google.com/url?q=https://leetcode.com/problems/flatten-binary-tree-to-linked-list&sa=D&source=editors&ust=1707482066874956&usg=AOvVaw0zbKglc29A_yGh8xhNOOTJ) | **Medium** |
| 85 | **Hash Table, String, Design, Trie** | [https://leetcode.com/problems/implement-trie-prefix-tree](https://www.google.com/url?q=https://leetcode.com/problems/implement-trie-prefix-tree&sa=D&source=editors&ust=1707482066875072&usg=AOvVaw0tj1Vhoj9_tLXOom9JXPrA) | **Medium** |
| 86 | **Array, Stack, Monotonic Stack** | [https://leetcode.com/problems/daily-temperatures](https://www.google.com/url?q=https://leetcode.com/problems/daily-temperatures&sa=D&source=editors&ust=1707482066875225&usg=AOvVaw00Q6a5aoS53rUgBJhlF9zP) | **Medium** |
| 87 | **Hash Table, Two Pointers, String, Greedy** | [https://leetcode.com/problems/partition-labels](https://www.google.com/url?q=https://leetcode.com/problems/partition-labels&sa=D&source=editors&ust=1707482066875341&usg=AOvVaw2c1I1FaZxYIHgVSM3SYmn2) | **Medium** |
| 88 | **Math, Dynamic Programming, Breadth-First Search** | [https://leetcode.com/problems/perfect-squares](https://www.google.com/url?q=https://leetcode.com/problems/perfect-squares&sa=D&source=editors&ust=1707482066875447&usg=AOvVaw2BIHk-QFSWtbenKJ7SFCIA) | **Medium** |
| 89 | **String, Dynamic Programming** | [https://leetcode.com/problems/decode-ways](https://www.google.com/url?q=https://leetcode.com/problems/decode-ways&sa=D&source=editors&ust=1707482066875585&usg=AOvVaw00JcU1Ony4qZz_Yb41AmwM) | **Medium** |
| 90 | **Two Pointers, Design, Sorting, Heap (Priority Queue), Data Stream** | [https://leetcode.com/problems/find-median-from-data-stream](https://www.google.com/url?q=https://leetcode.com/problems/find-median-from-data-stream&sa=D&source=editors&ust=1707482066875720&usg=AOvVaw2XSr45depIk8PhaiSSAwG6) | **Hard** |
| 91 | **Array, Dynamic Programming, Stack, Matrix, Monotonic Stack** | [https://leetcode.com/problems/maximal-rectangle](https://www.google.com/url?q=https://leetcode.com/problems/maximal-rectangle&sa=D&source=editors&ust=1707482066875835&usg=AOvVaw3DJwavW9KWS7Dbdc1SW-zD) | **Hard** |
| 92 | **Array, Hash Table, Greedy, Sorting, Heap (Priority Queue), Counting** | [https://leetcode.com/problems/task-scheduler](https://www.google.com/url?q=https://leetcode.com/problems/task-scheduler&sa=D&source=editors&ust=1707482066875979&usg=AOvVaw2SGEFZC_z0HzOGSTVxiORi) | **Medium** |
| 93 | **Linked List, Recursion** | [https://leetcode.com/problems/reverse-nodes-in-k-group](https://www.google.com/url?q=https://leetcode.com/problems/reverse-nodes-in-k-group&sa=D&source=editors&ust=1707482066876154&usg=AOvVaw2L-DMs8zX9aerJ1dxmuRMd) | **Hard** |
| 94 | **Array, Matrix, Simulation** | [https://leetcode.com/problems/spiral-matrix](https://www.google.com/url?q=https://leetcode.com/problems/spiral-matrix&sa=D&source=editors&ust=1707482066876335&usg=AOvVaw0GPUuLlAAh-fvBRUFmDzZ4) | **Medium** |
| 95 | **Array, Dynamic Programming, Backtracking** | [https://leetcode.com/problems/target-sum](https://www.google.com/url?q=https://leetcode.com/problems/target-sum&sa=D&source=editors&ust=1707482066876712&usg=AOvVaw3xGvBNTXD3c8Bo8cN___gK) | **Medium** |
| 96 | **Array, Hash Table** | [https://leetcode.com/problems/find-all-numbers-disappeared-in-an-array](https://www.google.com/url?q=https://leetcode.com/problems/find-all-numbers-disappeared-in-an-array&sa=D&source=editors&ust=1707482066876855&usg=AOvVaw2lUz2E4gdgVQRaq8XlXGEI) | **Easy** |
| 97 | **Depth-First Search, Breadth-First Search, Graph, Topological Sort** | [https://leetcode.com/problems/course-schedule-ii](https://www.google.com/url?q=https://leetcode.com/problems/course-schedule-ii&sa=D&source=editors&ust=1707482066876968&usg=AOvVaw3hZl98GRssSM_2_9gh7l6C) | **Medium** |
| 98 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/merge-two-binary-trees](https://www.google.com/url?q=https://leetcode.com/problems/merge-two-binary-trees&sa=D&source=editors&ust=1707482066877113&usg=AOvVaw0m4CUSM9HuJo4STHuYMN4y) | **Easy** |
| 99 | **String, Tree, Depth-First Search, Breadth-First Search, Design, Binary Tree** | [https://leetcode.com/problems/serialize-and-deserialize-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/serialize-and-deserialize-binary-tree&sa=D&source=editors&ust=1707482066877353&usg=AOvVaw1x6ZCMiRt1AdyZALqFWcoL) | **Hard** |
| 100 | **Hash Table, Linked List, Two Pointers** | [https://leetcode.com/problems/linked-list-cycle-ii](https://www.google.com/url?q=https://leetcode.com/problems/linked-list-cycle-ii&sa=D&source=editors&ust=1707482066877461&usg=AOvVaw30c6O1OSifuY1Bl1U4s93R) | **Medium** |
| 101 | **Dynamic Programming, Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/house-robber-iii](https://www.google.com/url?q=https://leetcode.com/problems/house-robber-iii&sa=D&source=editors&ust=1707482066877566&usg=AOvVaw0te-4cjsKPkMAGz0umDCHY) | **Medium** |
| 102 | **Linked List, Two Pointers, Divide and Conquer, Sorting, Merge Sort** | [https://leetcode.com/problems/sort-list](https://www.google.com/url?q=https://leetcode.com/problems/sort-list&sa=D&source=editors&ust=1707482066877667&usg=AOvVaw38B8b73rtCrJlO5eIHMxBt) | **Medium** |
| 103 | **Hash Table, String, Sliding Window** | [https://leetcode.com/problems/find-all-anagrams-in-a-string](https://www.google.com/url?q=https://leetcode.com/problems/find-all-anagrams-in-a-string&sa=D&source=editors&ust=1707482066877773&usg=AOvVaw2CP2YR9XNNIao_QyOIYoQc) | **Medium** |
| 104 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-right-side-view](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-right-side-view&sa=D&source=editors&ust=1707482066877891&usg=AOvVaw1i_grUShnpcAappyCenSj6) | **Medium** |
| 105 | **String, Dynamic Programming** | [https://leetcode.com/problems/palindromic-substrings](https://www.google.com/url?q=https://leetcode.com/problems/palindromic-substrings&sa=D&source=editors&ust=1707482066878011&usg=AOvVaw15kWL5y4a-8MUzqVnPATda) | **Medium** |
| 106 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/maximum-depth-of-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/maximum-depth-of-binary-tree&sa=D&source=editors&ust=1707482066878113&usg=AOvVaw2cYN7DFxtUAK82hvGjv6S9) | **Easy** |
| 107 | **Array, Breadth-First Search, Matrix** | [https://leetcode.com/problems/rotting-oranges](https://www.google.com/url?q=https://leetcode.com/problems/rotting-oranges&sa=D&source=editors&ust=1707482066878224&usg=AOvVaw1n1ctPaII9XfrwjlOwGONL) | **Medium** |
| 108 | **Array, Hash Table, Matrix** | [https://leetcode.com/problems/set-matrix-zeroes](https://www.google.com/url?q=https://leetcode.com/problems/set-matrix-zeroes&sa=D&source=editors&ust=1707482066878332&usg=AOvVaw1gP1Ah1bkn0wnzwt3PLQvF) | **Medium** |
| 109 | **Array, Divide and Conquer, Tree, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/convert-sorted-array-to-binary-search-tree](https://www.google.com/url?q=https://leetcode.com/problems/convert-sorted-array-to-binary-search-tree&sa=D&source=editors&ust=1707482066878441&usg=AOvVaw2PgkBb8QB8kJKhenRnGa7I) | **Easy** |
| 110 | **Array, Binary Search, Matrix** | [https://leetcode.com/problems/search-a-2d-matrix](https://www.google.com/url?q=https://leetcode.com/problems/search-a-2d-matrix&sa=D&source=editors&ust=1707482066878548&usg=AOvVaw2FGo6PufGjP2Gy7enO6MYk) | **Medium** |
| 111 | **Array, Binary Search** | [https://leetcode.com/problems/find-minimum-in-rotated-sorted-array](https://www.google.com/url?q=https://leetcode.com/problems/find-minimum-in-rotated-sorted-array&sa=D&source=editors&ust=1707482066878707&usg=AOvVaw0NsOf-jqWFsWtYHaf2nqEz) | **Medium** |
| 112 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/populating-next-right-pointers-in-each-node](https://www.google.com/url?q=https://leetcode.com/problems/populating-next-right-pointers-in-each-node&sa=D&source=editors&ust=1707482066878891&usg=AOvVaw1ji7reK7lDXHDIgKu--fTb) | **Medium** |
| 113 | **Array, Two Pointers** | [https://leetcode.com/problems/remove-duplicates-from-sorted-array](https://www.google.com/url?q=https://leetcode.com/problems/remove-duplicates-from-sorted-array&sa=D&source=editors&ust=1707482066879083&usg=AOvVaw34DUd2GwdZwA2pEdSZqfKZ) | **Easy** |
| 114 | **Tree, Depth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/kth-smallest-element-in-a-bst](https://www.google.com/url?q=https://leetcode.com/problems/kth-smallest-element-in-a-bst&sa=D&source=editors&ust=1707482066879261&usg=AOvVaw2XFD3Hta0jS9lpc_AD4DR-) | **Medium** |
| 115 | **Array, Binary Search, Sliding Window, Prefix Sum** | [https://leetcode.com/problems/minimum-size-subarray-sum](https://www.google.com/url?q=https://leetcode.com/problems/minimum-size-subarray-sum&sa=D&source=editors&ust=1707482066879425&usg=AOvVaw3G9ES1anAsgniEoo1k4d9g) | **Medium** |
| 116 | **Array, Hash Table** | [https://leetcode.com/problems/find-all-duplicates-in-an-array](https://www.google.com/url?q=https://leetcode.com/problems/find-all-duplicates-in-an-array&sa=D&source=editors&ust=1707482066879555&usg=AOvVaw2g_gOBQ5HgiWteju9Y4aJH) | **Medium** |
| 117 | **Array, Dynamic Programming** | [https://leetcode.com/problems/burst-balloons](https://www.google.com/url?q=https://leetcode.com/problems/burst-balloons&sa=D&source=editors&ust=1707482066879689&usg=AOvVaw2rR53AhiV-YW51crZeluF4) | **Hard** |
| 118 | **Dynamic Programming, Bit Manipulation** | [https://leetcode.com/problems/counting-bits](https://www.google.com/url?q=https://leetcode.com/problems/counting-bits&sa=D&source=editors&ust=1707482066879849&usg=AOvVaw2FmHfElh55fIPTIIgET5r1) | **Easy** |
| 119 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/all-nodes-distance-k-in-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/all-nodes-distance-k-in-binary-tree&sa=D&source=editors&ust=1707482066879966&usg=AOvVaw0QcBAkc-AHxfqRsRJ_Mgfl) | **Medium** |
| 120 | **Linked List, Recursion** | [https://leetcode.com/problems/swap-nodes-in-pairs](https://www.google.com/url?q=https://leetcode.com/problems/swap-nodes-in-pairs&sa=D&source=editors&ust=1707482066880112&usg=AOvVaw2eXO9dNVpQgB3wJsa3JdnC) | **Medium** |
| 121 | **Linked List** | [https://leetcode.com/problems/reverse-linked-list-ii](https://www.google.com/url?q=https://leetcode.com/problems/reverse-linked-list-ii&sa=D&source=editors&ust=1707482066880250&usg=AOvVaw2QfIp39trv1bLAeKcfZkr8) | **Medium** |
| 122 | **Array, String, Backtracking, Trie, Matrix** | [https://leetcode.com/problems/word-search-ii](https://www.google.com/url?q=https://leetcode.com/problems/word-search-ii&sa=D&source=editors&ust=1707482066880360&usg=AOvVaw1YXVDrXH4hcP9afoUrkzOd) | **Hard** |
| 123 | **String, Dynamic Programming** | [https://leetcode.com/problems/longest-common-subsequence](https://www.google.com/url?q=https://leetcode.com/problems/longest-common-subsequence&sa=D&source=editors&ust=1707482066880463&usg=AOvVaw3ido3qPfPuqCFrhvsriMXb) | **Medium** |
| 124 | **Array, Binary Search, Sorting, Heap (Priority Queue), Matrix** | [https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix](https://www.google.com/url?q=https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix&sa=D&source=editors&ust=1707482066880565&usg=AOvVaw0TU5RQfwVq_dmnc49SPZle) | **Medium** |
| 125 | **Array, Two Pointers, Sorting** | [https://leetcode.com/problems/4sum](https://www.google.com/url?q=https://leetcode.com/problems/4sum&sa=D&source=editors&ust=1707482066880668&usg=AOvVaw2WmW_ILsBAQc_yz2AyWOle) | **Medium** |
| 126 | **Array, Dynamic Programming** | [https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iii](https://www.google.com/url?q=https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iii&sa=D&source=editors&ust=1707482066880771&usg=AOvVaw0rhooMuMgoVSNNRvnPVyHD) | **Hard** |
| 127 | **Array, Dynamic Programming** | [https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-cooldown](https://www.google.com/url?q=https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-cooldown&sa=D&source=editors&ust=1707482066880882&usg=AOvVaw19GXqNM_Lrcf08M__mat-y) | **Medium** |
| 128 | **String, Dynamic Programming, Backtracking** | [https://leetcode.com/problems/palindrome-partitioning](https://www.google.com/url?q=https://leetcode.com/problems/palindrome-partitioning&sa=D&source=editors&ust=1707482066880988&usg=AOvVaw0b-xpeTT4MoGBXHhVME89k) | **Medium** |
| 129 | **Linked List, Two Pointers, Stack, Recursion** | [https://leetcode.com/problems/reorder-list](https://www.google.com/url?q=https://leetcode.com/problems/reorder-list&sa=D&source=editors&ust=1707482066881094&usg=AOvVaw3363uwK2FJgEeo7Qyha61U) | **Medium** |
| 130 | **Array, Dynamic Programming** | [https://leetcode.com/problems/min-cost-climbing-stairs](https://www.google.com/url?q=https://leetcode.com/problems/min-cost-climbing-stairs&sa=D&source=editors&ust=1707482066881197&usg=AOvVaw3gy5yoa7SZnvgU_mophqTV) | **Easy** |
| 131 | **Array, Depth-First Search, Breadth-First Search, Union Find, Matrix** | [https://leetcode.com/problems/max-area-of-island](https://www.google.com/url?q=https://leetcode.com/problems/max-area-of-island&sa=D&source=editors&ust=1707482066881310&usg=AOvVaw3b_OWjLn-gjwrWqquGl_go) | **Medium** |
| 132 | **Array, Binary Search, Divide and Conquer, Binary Indexed Tree, Segment Tree, Merge Sort, Ordered Set** | [https://leetcode.com/problems/count-of-smaller-numbers-after-self](https://www.google.com/url?q=https://leetcode.com/problems/count-of-smaller-numbers-after-self&sa=D&source=editors&ust=1707482066881422&usg=AOvVaw2wxNCKW-LfDb_qAeEbNOYi) | **Hard** |
| 133 | **Array, Two Pointers, Sorting** | [https://leetcode.com/problems/3sum-closest](https://www.google.com/url?q=https://leetcode.com/problems/3sum-closest&sa=D&source=editors&ust=1707482066881529&usg=AOvVaw0wos6eKayXZRxFI4FG6awp) | **Medium** |
| 134 | **Tree, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-zigzag-level-order-traversal](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-zigzag-level-order-traversal&sa=D&source=editors&ust=1707482066881630&usg=AOvVaw1_QYzOQhSn9IE3cKMsclbf) | **Medium** |
| 135 | **Array, Binary Search** | [https://leetcode.com/problems/find-peak-element](https://www.google.com/url?q=https://leetcode.com/problems/find-peak-element&sa=D&source=editors&ust=1707482066881771&usg=AOvVaw3yWLIRqJ3LFjFn6Kk2HqCZ) | **Medium** |
| 136 | **Array, Hash Table, Math, Design, Randomized** | [https://leetcode.com/problems/insert-delete-getrandom-o1](https://www.google.com/url?q=https://leetcode.com/problems/insert-delete-getrandom-o1&sa=D&source=editors&ust=1707482066881882&usg=AOvVaw0MYm7fGE5Ih-NP_z1JLCEv) | **Medium** |
| 137 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/balanced-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/balanced-binary-tree&sa=D&source=editors&ust=1707482066881986&usg=AOvVaw0IAfx7LEN5tSjSn5vMbjgM) | **Easy** |
| 138 | **Math** | [https://leetcode.com/problems/palindrome-number](https://www.google.com/url?q=https://leetcode.com/problems/palindrome-number&sa=D&source=editors&ust=1707482066882125&usg=AOvVaw0UapPdtNa51cogKs_MIWK3) | **Easy** |
| 139 | **Array, Backtracking** | [https://leetcode.com/problems/n-queens](https://www.google.com/url?q=https://leetcode.com/problems/n-queens&sa=D&source=editors&ust=1707482066882271&usg=AOvVaw2ECLCuKd9_UZ_RpDbq2tXe) | **Hard** |
| 140 | **Array, Math, Divide and Conquer, Geometry, Sorting, Heap (Priority Queue), Quickselect** | [https://leetcode.com/problems/k-closest-points-to-origin](https://www.google.com/url?q=https://leetcode.com/problems/k-closest-points-to-origin&sa=D&source=editors&ust=1707482066882390&usg=AOvVaw1oegqDHXn-l7XkAYVGIhuZ) | **Medium** |
| 141 | **Linked List** | [https://leetcode.com/problems/odd-even-linked-list](https://www.google.com/url?q=https://leetcode.com/problems/odd-even-linked-list&sa=D&source=editors&ust=1707482066882499&usg=AOvVaw2_KvZYz7k6nyV2EwvEt6rT) | **Medium** |
| 142 | **Depth-First Search, Breadth-First Search, Graph, Topological Sort** | [https://leetcode.com/problems/minimum-height-trees](https://www.google.com/url?q=https://leetcode.com/problems/minimum-height-trees&sa=D&source=editors&ust=1707482066882603&usg=AOvVaw2X84HQeDalzfgGLMrybmeQ) | **Medium** |
| 143 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/same-tree](https://www.google.com/url?q=https://leetcode.com/problems/same-tree&sa=D&source=editors&ust=1707482066882737&usg=AOvVaw0Flj7ZEvPmyXVYsX3LFJT1) | **Easy** |
| 144 | **Tree, Depth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-search-tree](https://www.google.com/url?q=https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-search-tree&sa=D&source=editors&ust=1707482066882915&usg=AOvVaw2mMC9m1Azyrs0xvwWbTz6-) | **Easy** |
| 145 | **Array, Two Pointers, Stack, Greedy, Sorting, Monotonic Stack** | [https://leetcode.com/problems/shortest-unsorted-continuous-subarray](https://www.google.com/url?q=https://leetcode.com/problems/shortest-unsorted-continuous-subarray&sa=D&source=editors&ust=1707482066883097&usg=AOvVaw1PrwuNqd6iqMg2lOMeAFn1) | **Medium** |
| 146 | **Stack, Tree, Design, Binary Search Tree, Binary Tree, Iterator** | [https://leetcode.com/problems/binary-search-tree-iterator](https://www.google.com/url?q=https://leetcode.com/problems/binary-search-tree-iterator&sa=D&source=editors&ust=1707482066883269&usg=AOvVaw0uaMy9cnxt1wVtwv5aA-FH) | **Medium** |
| 147 | **Dynamic Programming, Depth-First Search, Breadth-First Search, Graph, Topological Sort, Memoization** | [https://leetcode.com/problems/longest-increasing-path-in-a-matrix](https://www.google.com/url?q=https://leetcode.com/problems/longest-increasing-path-in-a-matrix&sa=D&source=editors&ust=1707482066883462&usg=AOvVaw3Bb6MJ16QOihJ4QgWX2Von) | **Hard** |
| 148 | **Array, Greedy, Sorting** | [https://leetcode.com/problems/queue-reconstruction-by-height](https://www.google.com/url?q=https://leetcode.com/problems/queue-reconstruction-by-height&sa=D&source=editors&ust=1707482066883633&usg=AOvVaw0UHY_dBp_-WU10bZ5cSMGM) | **Medium** |
| 149 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/path-sum](https://www.google.com/url?q=https://leetcode.com/problems/path-sum&sa=D&source=editors&ust=1707482066883779&usg=AOvVaw0DHeKzxT9uBHRmw1k1MF_4) | **Easy** |
| 150 | **Tree, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/delete-node-in-a-bst](https://www.google.com/url?q=https://leetcode.com/problems/delete-node-in-a-bst&sa=D&source=editors&ust=1707482066883967&usg=AOvVaw1T5s7UbhWq90uXaqqbq0es) | **Medium** |
| 151 | **Tree, Depth-First Search, String Matching, Binary Tree, Hash Function** | [https://leetcode.com/problems/subtree-of-another-tree](https://www.google.com/url?q=https://leetcode.com/problems/subtree-of-another-tree&sa=D&source=editors&ust=1707482066884165&usg=AOvVaw1hGOmlrL34R76Du0ApCmQ7) | **Easy** |
| 152 | **Array, Hash Table, Math, Bit Manipulation, Sorting** | [https://leetcode.com/problems/missing-number](https://www.google.com/url?q=https://leetcode.com/problems/missing-number&sa=D&source=editors&ust=1707482066884351&usg=AOvVaw04rT_xU3Ljm8nwtUAH98NF) | **Easy** |
| 153 | **String, Dynamic Programming** | [https://leetcode.com/problems/longest-palindromic-subsequence](https://www.google.com/url?q=https://leetcode.com/problems/longest-palindromic-subsequence&sa=D&source=editors&ust=1707482066884530&usg=AOvVaw1e35qCyMRy2n7XgS5IAWsP) | **Medium** |
| 154 | **Array, Depth-First Search, Breadth-First Search, Union Find, Graph, Shortest Path** | [https://leetcode.com/problems/evaluate-division](https://www.google.com/url?q=https://leetcode.com/problems/evaluate-division&sa=D&source=editors&ust=1707482066884720&usg=AOvVaw3EnytqG2kJKFQYYxzZMb0Y) | **Medium** |
| 155 | **Array, Dynamic Programming** | [https://leetcode.com/problems/house-robber-ii](https://www.google.com/url?q=https://leetcode.com/problems/house-robber-ii&sa=D&source=editors&ust=1707482066884882&usg=AOvVaw3GVJsNqBPMW3_ii4puSsyh) | **Medium** |
| 156 | **Array, Dynamic Programming** | [https://leetcode.com/problems/pascals-triangle](https://www.google.com/url?q=https://leetcode.com/problems/pascals-triangle&sa=D&source=editors&ust=1707482066885012&usg=AOvVaw0oyXtNVOC1GXmeEuYsYuEf) | **Easy** |
| 157 | **Hash Table, Depth-First Search, Breadth-First Search, Graph** | [https://leetcode.com/problems/clone-graph](https://www.google.com/url?q=https://leetcode.com/problems/clone-graph&sa=D&source=editors&ust=1707482066885119&usg=AOvVaw3J6l2eS6v5Uka7SgdPdJ15) | **Medium** |
| 158 | **Array, Dynamic Programming** | [https://leetcode.com/problems/coin-change-2](https://www.google.com/url?q=https://leetcode.com/problems/coin-change-2&sa=D&source=editors&ust=1707482066885238&usg=AOvVaw1cvUgNu6gZzi5MQRMn19Pv) | **Medium** |
| 159 | **Hash Table, Math, Two Pointers** | [https://leetcode.com/problems/happy-number](https://www.google.com/url?q=https://leetcode.com/problems/happy-number&sa=D&source=editors&ust=1707482066885385&usg=AOvVaw1YWcpfF_nr74bONTXIvl5_) | **Easy** |
| 160 | **Depth-First Search, Breadth-First Search, Union Find, Graph** | [https://leetcode.com/problems/number-of-provinces](https://www.google.com/url?q=https://leetcode.com/problems/number-of-provinces&sa=D&source=editors&ust=1707482066885510&usg=AOvVaw0GOIS61U9PTDTNE6WXEur7) | **Medium** |
| 161 | **Array, Binary Search** | [https://leetcode.com/problems/single-element-in-a-sorted-array](https://www.google.com/url?q=https://leetcode.com/problems/single-element-in-a-sorted-array&sa=D&source=editors&ust=1707482066885665&usg=AOvVaw1FgukwMdU_5FOzgynMQ8l-) | **Medium** |
| 162 | **Array, Greedy** | [https://leetcode.com/problems/gas-station](https://www.google.com/url?q=https://leetcode.com/problems/gas-station&sa=D&source=editors&ust=1707482066885808&usg=AOvVaw3TH83_zqMyv6D2OVbGXGRw) | **Medium** |
| 163 | **Dynamic Programming, Backtracking, Tree, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/unique-binary-search-trees-ii](https://www.google.com/url?q=https://leetcode.com/problems/unique-binary-search-trees-ii&sa=D&source=editors&ust=1707482066885997&usg=AOvVaw0dXIO3ks-LAXf6_Gs6leJo) | **Medium** |
| 164 | **Dynamic Programming, Depth-First Search, Breadth-First Search, Graph, Heap (Priority Queue), Shortest Path** | [https://leetcode.com/problems/cheapest-flights-within-k-stops](https://www.google.com/url?q=https://leetcode.com/problems/cheapest-flights-within-k-stops&sa=D&source=editors&ust=1707482066886174&usg=AOvVaw1cs5DGScZkwIpi199FnupK) | **Medium** |
| 165 | **Linked List, Two Pointers** | [https://leetcode.com/problems/middle-of-the-linked-list](https://www.google.com/url?q=https://leetcode.com/problems/middle-of-the-linked-list&sa=D&source=editors&ust=1707482066886351&usg=AOvVaw2dm1eFe-vn4VrRb6UAt_V3) | **Easy** |
| 166 | **String, Stack, Greedy, Monotonic Stack** | [https://leetcode.com/problems/remove-k-digits](https://www.google.com/url?q=https://leetcode.com/problems/remove-k-digits&sa=D&source=editors&ust=1707482066886529&usg=AOvVaw3vbf9JGvv84fFzoafe-9eY) | **Medium** |
| 167 | **Array, Backtracking, Matrix** | [https://leetcode.com/problems/sudoku-solver](https://www.google.com/url?q=https://leetcode.com/problems/sudoku-solver&sa=D&source=editors&ust=1707482066886719&usg=AOvVaw3ejjOKdZr1IUBQgDaby4su) | **Hard** |
| 168 | **Array, Depth-First Search, Breadth-First Search, Union Find, Matrix** | [https://leetcode.com/problems/surrounded-regions](https://www.google.com/url?q=https://leetcode.com/problems/surrounded-regions&sa=D&source=editors&ust=1707482066886886&usg=AOvVaw2gbsiG_WLrubXifl3Hvk9u) | **Medium** |
| 169 | **Binary Search, Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/count-complete-tree-nodes](https://www.google.com/url?q=https://leetcode.com/problems/count-complete-tree-nodes&sa=D&source=editors&ust=1707482066886991&usg=AOvVaw3ji9pN1HuJB5SIg9zAADwH) | **Medium** |
| 170 | **String, Backtracking, Breadth-First Search** | [https://leetcode.com/problems/remove-invalid-parentheses](https://www.google.com/url?q=https://leetcode.com/problems/remove-invalid-parentheses&sa=D&source=editors&ust=1707482066887096&usg=AOvVaw3J9REir_pGCYm5sNSpANE-) | **Hard** |
| 171 | **Array, Dynamic Programming, Backtracking, Bit Manipulation, Memoization, Bitmask** | [https://leetcode.com/problems/partition-to-k-equal-sum-subsets](https://www.google.com/url?q=https://leetcode.com/problems/partition-to-k-equal-sum-subsets&sa=D&source=editors&ust=1707482066887199&usg=AOvVaw1vLWz5Z3Unb_oI-JkHR9j0) | **Medium** |
| 172 | **Array, Two Pointers, Binary Search** | [https://leetcode.com/problems/two-sum-ii-input-array-is-sorted](https://www.google.com/url?q=https://leetcode.com/problems/two-sum-ii-input-array-is-sorted&sa=D&source=editors&ust=1707482066887335&usg=AOvVaw13r0mKWnVcDZuN0GFk_0zE) | **Easy** |
| 173 | **Hash Table, String, Queue, Counting** | [https://leetcode.com/problems/first-unique-character-in-a-string](https://www.google.com/url?q=https://leetcode.com/problems/first-unique-character-in-a-string&sa=D&source=editors&ust=1707482066887496&usg=AOvVaw1R_khBWlXs4M8kAgnBeWsB) | **Easy** |
| 174 | **String, Greedy, Sorting** | [https://leetcode.com/problems/largest-number](https://www.google.com/url?q=https://leetcode.com/problems/largest-number&sa=D&source=editors&ust=1707482066887691&usg=AOvVaw3IzdKmwKCb2yS3jfUkBqu-) | **Medium** |
| 175 | **Array, Math, Enumeration, Number Theory** | [https://leetcode.com/problems/count-primes](https://www.google.com/url?q=https://leetcode.com/problems/count-primes&sa=D&source=editors&ust=1707482066887865&usg=AOvVaw2y1vOEJwKcc59VWFhHf_4I) | **Medium** |
| 176 | **Array, Two Pointers, Sorting** | [https://leetcode.com/problems/squares-of-a-sorted-array](https://www.google.com/url?q=https://leetcode.com/problems/squares-of-a-sorted-array&sa=D&source=editors&ust=1707482066888045&usg=AOvVaw3zj-4kNSAuLLtWPjaQFY-F) | **Easy** |
| 177 | **Array, Backtracking** | [https://leetcode.com/problems/permutations-ii](https://www.google.com/url?q=https://leetcode.com/problems/permutations-ii&sa=D&source=editors&ust=1707482066888216&usg=AOvVaw34ylwxSiMQlzr0-C0dxO5x) | **Medium** |
| 178 | **Hash Table, String, Dynamic Programming, Backtracking, Trie, Memoization** | [https://leetcode.com/problems/word-break-ii](https://www.google.com/url?q=https://leetcode.com/problems/word-break-ii&sa=D&source=editors&ust=1707482066888412&usg=AOvVaw3Xr5_o4pXUX-SPjtJMxzip) | **Hard** |
| 179 | **Array, Backtracking** | [https://leetcode.com/problems/combination-sum-ii](https://www.google.com/url?q=https://leetcode.com/problems/combination-sum-ii&sa=D&source=editors&ust=1707482066888607&usg=AOvVaw0fMcSeuSy7EeXooZFmwV84) | **Medium** |
| 180 | **Array, Dynamic Programming** | [https://leetcode.com/problems/triangle](https://www.google.com/url?q=https://leetcode.com/problems/triangle&sa=D&source=editors&ust=1707482066888813&usg=AOvVaw2cORtnqV-t30dWgxUV0xAQ) | **Medium** |
| 181 | **Linked List, Recursion** | [https://leetcode.com/problems/remove-linked-list-elements](https://www.google.com/url?q=https://leetcode.com/problems/remove-linked-list-elements&sa=D&source=editors&ust=1707482066888964&usg=AOvVaw268b8zlPlKlL6Lj7sa1tuf) | **Easy** |
| 182 | **Linked List, Two Pointers** | [https://leetcode.com/problems/remove-duplicates-from-sorted-list-ii](https://www.google.com/url?q=https://leetcode.com/problems/remove-duplicates-from-sorted-list-ii&sa=D&source=editors&ust=1707482066889115&usg=AOvVaw3UQojbN_0yjbnLdf-n6pYJ) | **Medium** |
| 183 | **Array, Hash Table, Matrix** | [https://leetcode.com/problems/valid-sudoku](https://www.google.com/url?q=https://leetcode.com/problems/valid-sudoku&sa=D&source=editors&ust=1707482066889229&usg=AOvVaw3jH6eg7zE0VAkIWHE57tG0) | **Medium** |
| 184 | **Backtracking, Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/path-sum-ii](https://www.google.com/url?q=https://leetcode.com/problems/path-sum-ii&sa=D&source=editors&ust=1707482066889409&usg=AOvVaw3yuz4n0fvFgM_qhFp8X9Xw) | **Medium** |
| 185 | **Linked List, Divide and Conquer, Tree, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/convert-sorted-list-to-binary-search-tree](https://www.google.com/url?q=https://leetcode.com/problems/convert-sorted-list-to-binary-search-tree&sa=D&source=editors&ust=1707482066889657&usg=AOvVaw3hiAtDDCkWQZzYyMkAAHPv) | **Medium** |
| 186 | **Array** | [https://leetcode.com/problems/insert-interval](https://www.google.com/url?q=https://leetcode.com/problems/insert-interval&sa=D&source=editors&ust=1707482066889837&usg=AOvVaw1Hh03GvxiAIo03ZT0Cne6b) | **Medium** |
| 187 | **String, Dynamic Programming, Greedy, Recursion** | [https://leetcode.com/problems/wildcard-matching](https://www.google.com/url?q=https://leetcode.com/problems/wildcard-matching&sa=D&source=editors&ust=1707482066889944&usg=AOvVaw2ePS3tNH2brSaPAiR57Xr7) | **Hard** |
| 188 | **Array, Backtracking, Bit Manipulation** | [https://leetcode.com/problems/subsets-ii](https://www.google.com/url?q=https://leetcode.com/problems/subsets-ii&sa=D&source=editors&ust=1707482066890050&usg=AOvVaw0K0v-r5njr8lboQaYzz9TD) | **Medium** |
| 189 | **Array, Dynamic Programming, Matrix** | [https://leetcode.com/problems/unique-paths-ii](https://www.google.com/url?q=https://leetcode.com/problems/unique-paths-ii&sa=D&source=editors&ust=1707482066890161&usg=AOvVaw1cIBjs2SDvd8FdoJCP1Y9l) | **Medium** |
| 190 | **Array, Binary Search, Dynamic Programming, Greedy** | [https://leetcode.com/problems/split-array-largest-sum](https://www.google.com/url?q=https://leetcode.com/problems/split-array-largest-sum&sa=D&source=editors&ust=1707482066890329&usg=AOvVaw2wGrX_TD20TJMDnJ-x28xi) | **Hard** |
| 191 | **Hash Table, String, Greedy, Sorting, Heap (Priority Queue), Counting** | [https://leetcode.com/problems/reorganize-string](https://www.google.com/url?q=https://leetcode.com/problems/reorganize-string&sa=D&source=editors&ust=1707482066890518&usg=AOvVaw1izpHZzex3f78QW2tfLic5) | **Medium** |
| 192 | **Hash Table, Two Pointers, String, Sliding Window** | [https://leetcode.com/problems/permutation-in-string](https://www.google.com/url?q=https://leetcode.com/problems/permutation-in-string&sa=D&source=editors&ust=1707482066890691&usg=AOvVaw3-x0jaalrZxH2OAYahzuFV) | **Medium** |
| 193 | **Array, Hash Table, Sorting, Counting** | [https://leetcode.com/problems/majority-element-ii](https://www.google.com/url?q=https://leetcode.com/problems/majority-element-ii&sa=D&source=editors&ust=1707482066890837&usg=AOvVaw29GJd2CXf9xlMlyclabt0e) | **Medium** |
| 194 | **Array, Hash Table, Divide and Conquer, Tree, Binary Tree** | [https://leetcode.com/problems/construct-binary-tree-from-inorder-and-postorder-traversal](https://www.google.com/url?q=https://leetcode.com/problems/construct-binary-tree-from-inorder-and-postorder-traversal&sa=D&source=editors&ust=1707482066890993&usg=AOvVaw2uTrGdtl5vC3X0b4SOA5QF) | **Medium** |
| 195 | **Array, Dynamic Programming** | [https://leetcode.com/problems/minimum-cost-for-tickets](https://www.google.com/url?q=https://leetcode.com/problems/minimum-cost-for-tickets&sa=D&source=editors&ust=1707482066891141&usg=AOvVaw3V63WrZouy-oiVduXeDVxX) | **Medium** |
| 196 | **Hash Table, String, Trie, Sorting, Heap (Priority Queue), Bucket Sort, Counting** | [https://leetcode.com/problems/top-k-frequent-words](https://www.google.com/url?q=https://leetcode.com/problems/top-k-frequent-words&sa=D&source=editors&ust=1707482066891284&usg=AOvVaw21qmqQCxs408dapes8YT0s) | **Medium** |
| 197 | **Array, Depth-First Search, Breadth-First Search, Matrix** | [https://leetcode.com/problems/island-perimeter](https://www.google.com/url?q=https://leetcode.com/problems/island-perimeter&sa=D&source=editors&ust=1707482066891438&usg=AOvVaw1OeLis7KlvjRWT08s7owOi) | **Easy** |
| 198 | **Linked List** | [https://leetcode.com/problems/remove-duplicates-from-sorted-list](https://www.google.com/url?q=https://leetcode.com/problems/remove-duplicates-from-sorted-list&sa=D&source=editors&ust=1707482066891582&usg=AOvVaw1EWS8x6d4-VX4DFpKmZ3ZD) | **Easy** |
| 199 | **Math, String, Bit Manipulation, Simulation** | [https://leetcode.com/problems/add-binary](https://www.google.com/url?q=https://leetcode.com/problems/add-binary&sa=D&source=editors&ust=1707482066891723&usg=AOvVaw0JNGQ1zJ4vFISoOt0NkbaI) | **Easy** |
| 200 | **Array, Dynamic Programming, Breadth-First Search, Matrix** | [https://leetcode.com/problems/01-matrix](https://www.google.com/url?q=https://leetcode.com/problems/01-matrix&sa=D&source=editors&ust=1707482066891879&usg=AOvVaw0EMDO7CTeSaahaS8WL3a27) | **Medium** |
| 201 | **Hash Table, String, Sorting** | [https://leetcode.com/problems/valid-anagram](https://www.google.com/url?q=https://leetcode.com/problems/valid-anagram&sa=D&source=editors&ust=1707482066892020&usg=AOvVaw22U-1PXrlXp6zLasmDHD3R) | **Easy** |
| 202 | **Math, String, Simulation** | [https://leetcode.com/problems/multiply-strings](https://www.google.com/url?q=https://leetcode.com/problems/multiply-strings&sa=D&source=editors&ust=1707482066892160&usg=AOvVaw2dx7kRetK5HPsyueavteS0) | **Medium** |
| 203 | **Binary Search, Interactive** | [https://leetcode.com/problems/first-bad-version](https://www.google.com/url?q=https://leetcode.com/problems/first-bad-version&sa=D&source=editors&ust=1707482066892368&usg=AOvVaw3k7X9AeXv0b9bELiSCLjaI) | **Easy** |
| 204 | **String, Depth-First Search, Design, Trie** | [https://leetcode.com/problems/design-add-and-search-words-data-structure](https://www.google.com/url?q=https://leetcode.com/problems/design-add-and-search-words-data-structure&sa=D&source=editors&ust=1707482066892550&usg=AOvVaw1KrAIygMGDUsD3rBUlBAWP) | **Medium** |
| 205 | **Array, Stack, Monotonic Stack** | [https://leetcode.com/problems/next-greater-element-ii](https://www.google.com/url?q=https://leetcode.com/problems/next-greater-element-ii&sa=D&source=editors&ust=1707482066892712&usg=AOvVaw3dZM_ubXPPnkTxtUnJohLh) | **Medium** |
| 206 | **Math, String, Stack** | [https://leetcode.com/problems/basic-calculator-ii](https://www.google.com/url?q=https://leetcode.com/problems/basic-calculator-ii&sa=D&source=editors&ust=1707482066892902&usg=AOvVaw2hSf1QqjuaZj7MLf5r4Tbo) | **Medium** |
| 207 | **Array, String, Depth-First Search, Breadth-First Search, Union Find** | [https://leetcode.com/problems/accounts-merge](https://www.google.com/url?q=https://leetcode.com/problems/accounts-merge&sa=D&source=editors&ust=1707482066893070&usg=AOvVaw3MqxEN5W8yYlweywH7wWVy) | **Medium** |
| 208 | **Two Pointers, String, Greedy** | [https://leetcode.com/problems/valid-palindrome-ii](https://www.google.com/url?q=https://leetcode.com/problems/valid-palindrome-ii&sa=D&source=editors&ust=1707482066893222&usg=AOvVaw0MbaNBcY7y8y5lOTcw5Gub) | **Easy** |
| 209 | **Linked List** | [https://leetcode.com/problems/delete-node-in-a-linked-list](https://www.google.com/url?q=https://leetcode.com/problems/delete-node-in-a-linked-list&sa=D&source=editors&ust=1707482066893421&usg=AOvVaw3EZyYSPZLNThiSLS2CV5v7) | **Easy** |
| 210 | **Array, Two Pointers** | [https://leetcode.com/problems/interval-list-intersections](https://www.google.com/url?q=https://leetcode.com/problems/interval-list-intersections&sa=D&source=editors&ust=1707482066893616&usg=AOvVaw1NDqTfI9F71_e_pp5ZwTYb) | **Medium** |
| 211 | **Linked List, Two Pointers** | [https://leetcode.com/problems/rotate-list](https://www.google.com/url?q=https://leetcode.com/problems/rotate-list&sa=D&source=editors&ust=1707482066893791&usg=AOvVaw1TYpL_yMbyzhAMwhwqOuGh) | **Medium** |
| 212 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/sum-root-to-leaf-numbers](https://www.google.com/url?q=https://leetcode.com/problems/sum-root-to-leaf-numbers&sa=D&source=editors&ust=1707482066893974&usg=AOvVaw1VaN5q6RiHWzhfAoStVMTN) | **Medium** |
| 213 | **Array, Dynamic Programming, Matrix** | [https://leetcode.com/problems/dungeon-game](https://www.google.com/url?q=https://leetcode.com/problems/dungeon-game&sa=D&source=editors&ust=1707482066894159&usg=AOvVaw0Kkzq7cG4RPvDEMNL0_231) | **Hard** |
| 214 | **Math, Recursion** | [https://leetcode.com/problems/powx-n](https://www.google.com/url?q=https://leetcode.com/problems/powx-n&sa=D&source=editors&ust=1707482066894323&usg=AOvVaw2ODcyqEoHZv9qdY_2H79A7) | **Medium** |
| 215 | **Array, Binary Search, Sliding Window, Prefix Sum** | [https://leetcode.com/problems/max-consecutive-ones-iii](https://www.google.com/url?q=https://leetcode.com/problems/max-consecutive-ones-iii&sa=D&source=editors&ust=1707482066894506&usg=AOvVaw1CqNcAfO6zQ2_8p0KpVGOE) | **Medium** |
| 216 | **Array, Hash Table, Two Pointers, Binary Search, Sorting** | [https://leetcode.com/problems/intersection-of-two-arrays-ii](https://www.google.com/url?q=https://leetcode.com/problems/intersection-of-two-arrays-ii&sa=D&source=editors&ust=1707482066894688&usg=AOvVaw22fvgh2NfpvwNRgULuG7kT) | **Easy** |
| 217 | **String, Stack, Greedy, Monotonic Stack** | [https://leetcode.com/problems/remove-duplicate-letters](https://www.google.com/url?q=https://leetcode.com/problems/remove-duplicate-letters&sa=D&source=editors&ust=1707482066894886&usg=AOvVaw3jZ01Db5KfjCH0pHk4pXgE) | **Medium** |
| 218 | **Tree, Depth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/recover-binary-search-tree](https://www.google.com/url?q=https://leetcode.com/problems/recover-binary-search-tree&sa=D&source=editors&ust=1707482066895058&usg=AOvVaw2ZMFlwqtyVyOzh_5bzLAHz) | **Medium** |
| 219 | **Depth-First Search, Breadth-First Search, Union Find, Graph** | [https://leetcode.com/problems/is-graph-bipartite](https://www.google.com/url?q=https://leetcode.com/problems/is-graph-bipartite&sa=D&source=editors&ust=1707482066895221&usg=AOvVaw3JUpTafyFx_23cEMmJ_nIu) | **Medium** |
| 220 | **Array, Hash Table, Prefix Sum** | [https://leetcode.com/problems/contiguous-array](https://www.google.com/url?q=https://leetcode.com/problems/contiguous-array&sa=D&source=editors&ust=1707482066895388&usg=AOvVaw2cZ2lY5f1xGSs7wbQF-eRV) | **Medium** |
| 221 | **Array, Two Pointers, Binary Search, Sorting, Heap (Priority Queue)** | [https://leetcode.com/problems/find-k-closest-elements](https://www.google.com/url?q=https://leetcode.com/problems/find-k-closest-elements&sa=D&source=editors&ust=1707482066895576&usg=AOvVaw2tjMdzKVGRjSfOW--BosCc) | **Medium** |
| 222 | **Two Pointers, String, Recursion** | [https://leetcode.com/problems/reverse-string](https://www.google.com/url?q=https://leetcode.com/problems/reverse-string&sa=D&source=editors&ust=1707482066895756&usg=AOvVaw0qGOr_4ShEkJZB91JYzMDt) | **Easy** |
| 223 | **Tree, Depth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/range-sum-of-bst](https://www.google.com/url?q=https://leetcode.com/problems/range-sum-of-bst&sa=D&source=editors&ust=1707482066895957&usg=AOvVaw1YTvQ-FB1DGThuXVNjdoHK) | **Easy** |
| 224 | **Hash Table, String, Sorting, Heap (Priority Queue), Bucket Sort, Counting** | [https://leetcode.com/problems/sort-characters-by-frequency](https://www.google.com/url?q=https://leetcode.com/problems/sort-characters-by-frequency&sa=D&source=editors&ust=1707482066896146&usg=AOvVaw3HtDHX7EHpEXGDg_SQV9ep) | **Medium** |
| 225 | **Hash Table, String, Sliding Window** | [https://leetcode.com/problems/longest-repeating-character-replacement](https://www.google.com/url?q=https://leetcode.com/problems/longest-repeating-character-replacement&sa=D&source=editors&ust=1707482066896340&usg=AOvVaw1S3_oTddxFadowF6aoXrwl) | **Medium** |
| 226 | **Stack, Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-postorder-traversal](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-postorder-traversal&sa=D&source=editors&ust=1707482066896516&usg=AOvVaw2yW2WOAaQc92pQHQAkEcDS) | **Easy** |
| 227 | **Array, Bit Manipulation** | [https://leetcode.com/problems/single-number-ii](https://www.google.com/url?q=https://leetcode.com/problems/single-number-ii&sa=D&source=editors&ust=1707482066896695&usg=AOvVaw2-ThSBXD0BWrArdqQllpOZ) | **Medium** |
| 228 | **Hash Table, Math, Dynamic Programming, Heap (Priority Queue)** | [https://leetcode.com/problems/ugly-number-ii](https://www.google.com/url?q=https://leetcode.com/problems/ugly-number-ii&sa=D&source=editors&ust=1707482066896890&usg=AOvVaw1az6g9QJYr3sxwEW9oK04X) | **Medium** |
| 229 | **Depth-First Search, Graph, Eulerian Circuit** | [https://leetcode.com/problems/reconstruct-itinerary](https://www.google.com/url?q=https://leetcode.com/problems/reconstruct-itinerary&sa=D&source=editors&ust=1707482066897086&usg=AOvVaw0E755CqeajDKYMc3fbNYC0) | **Hard** |
| 230 | **Array, Divide and Conquer, Binary Indexed Tree, Segment Tree, Line Sweep, Heap (Priority Queue), Ordered Set** | [https://leetcode.com/problems/the-skyline-problem](https://www.google.com/url?q=https://leetcode.com/problems/the-skyline-problem&sa=D&source=editors&ust=1707482066897279&usg=AOvVaw0cQn2wcko22kP_JaHAUCDo) | **Hard** |
| 231 | **Array, Bit Manipulation** | [https://leetcode.com/problems/single-number-iii](https://www.google.com/url?q=https://leetcode.com/problems/single-number-iii&sa=D&source=editors&ust=1707482066897473&usg=AOvVaw2eYGCTupKErDSjQ8O-Mf97) | **Medium** |
| 232 | **Two Pointers, String, Dynamic Programming** | [https://leetcode.com/problems/is-subsequence](https://www.google.com/url?q=https://leetcode.com/problems/is-subsequence&sa=D&source=editors&ust=1707482066897578&usg=AOvVaw1rJFSD-STHfAJS8lkIwZWK) | **Easy** |
| 233 | **Depth-First Search, Breadth-First Search, Graph, Heap (Priority Queue), Shortest Path** | [https://leetcode.com/problems/network-delay-time](https://www.google.com/url?q=https://leetcode.com/problems/network-delay-time&sa=D&source=editors&ust=1707482066897685&usg=AOvVaw16S7u1X7JNT9_jPRwfpwzD) | **Medium** |
| 234 | **String, Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-paths](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-paths&sa=D&source=editors&ust=1707482066897802&usg=AOvVaw0tDQIAwvs-Wt9Y8zG1WYid) | **Easy** |
| 235 | **Hash Table, String, Divide and Conquer, Sliding Window** | [https://leetcode.com/problems/longest-substring-with-at-least-k-repeating-characters](https://www.google.com/url?q=https://leetcode.com/problems/longest-substring-with-at-least-k-repeating-characters&sa=D&source=editors&ust=1707482066897934&usg=AOvVaw3CA_U441C1lH2BX7PPsPnA) | **Medium** |
| 236 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/minimum-depth-of-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/minimum-depth-of-binary-tree&sa=D&source=editors&ust=1707482066898044&usg=AOvVaw2yM8PxF8XjnChF6CrWZqUc) | **Easy** |
| 237 | **Array, Dynamic Programming, Greedy** | [https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-transaction-fee](https://www.google.com/url?q=https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-transaction-fee&sa=D&source=editors&ust=1707482066898155&usg=AOvVaw1IYW7Tj3uotyTEM-lfvzKD) | **Medium** |
| 238 | **Array** | [https://leetcode.com/problems/non-decreasing-array](https://www.google.com/url?q=https://leetcode.com/problems/non-decreasing-array&sa=D&source=editors&ust=1707482066898337&usg=AOvVaw3GPf5gyvcJ15r9a2VayFfy) | **Medium** |
| 239 | **String, Dynamic Programming** | [https://leetcode.com/problems/interleaving-string](https://www.google.com/url?q=https://leetcode.com/problems/interleaving-string&sa=D&source=editors&ust=1707482066898500&usg=AOvVaw1s_FpCCkkmogrLZrI3ckP8) | **Medium** |
| 240 | **Array, Matrix, Simulation** | [https://leetcode.com/problems/game-of-life](https://www.google.com/url?q=https://leetcode.com/problems/game-of-life&sa=D&source=editors&ust=1707482066898608&usg=AOvVaw1hoQ5zJjqFq0wMG5Oa7a0F) | **Medium** |
| 241 | **Array, Backtracking** | [https://leetcode.com/problems/combinations](https://www.google.com/url?q=https://leetcode.com/problems/combinations&sa=D&source=editors&ust=1707482066898712&usg=AOvVaw3xFurD0Gda5QkuiGyfMDbn) | **Medium** |
| 242 | **Stack, Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-preorder-traversal](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-preorder-traversal&sa=D&source=editors&ust=1707482066898821&usg=AOvVaw3BAwXeBAeAF-GhNgeYiCeG) | **Easy** |
| 243 | **Backtracking, Depth-First Search, Breadth-First Search, Graph** | [https://leetcode.com/problems/all-paths-from-source-to-target](https://www.google.com/url?q=https://leetcode.com/problems/all-paths-from-source-to-target&sa=D&source=editors&ust=1707482066899009&usg=AOvVaw0K_IgyB8Kb7wM94sla8B4A) | **Medium** |
| 244 | **Array, Math** | [https://leetcode.com/problems/plus-one](https://www.google.com/url?q=https://leetcode.com/problems/plus-one&sa=D&source=editors&ust=1707482066899134&usg=AOvVaw2LwoEzRndeH6yCgKjZIw1J) | **Easy** |
| 245 | **Array, Dynamic Programming** | [https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iv](https://www.google.com/url?q=https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iv&sa=D&source=editors&ust=1707482066899242&usg=AOvVaw1SoKagYAFdwiX2Uq5SKCXp) | **Hard** |
| 246 | **Array, Greedy** | [https://leetcode.com/problems/increasing-triplet-subsequence](https://www.google.com/url?q=https://leetcode.com/problems/increasing-triplet-subsequence&sa=D&source=editors&ust=1707482066899388&usg=AOvVaw0syPWCorScht-6gjzV5TZ5) | **Medium** |
| 247 | **Tree, Depth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/trim-a-binary-search-tree](https://www.google.com/url?q=https://leetcode.com/problems/trim-a-binary-search-tree&sa=D&source=editors&ust=1707482066899514&usg=AOvVaw1x4JFfsdNGW1IwJI1gOXwT) | **Medium** |
| 248 | **Two Pointers, String, Stack, Simulation** | [https://leetcode.com/problems/backspace-string-compare](https://www.google.com/url?q=https://leetcode.com/problems/backspace-string-compare&sa=D&source=editors&ust=1707482066899621&usg=AOvVaw178sVNHbJZ-y0tmZIbq93a) | **Easy** |
| 249 | **Array, Binary Search, Dynamic Programming, Sliding Window, Rolling Hash, Hash Function** | [https://leetcode.com/problems/maximum-length-of-repeated-subarray](https://www.google.com/url?q=https://leetcode.com/problems/maximum-length-of-repeated-subarray&sa=D&source=editors&ust=1707482066899764&usg=AOvVaw1_uP_PZaKuIGUPfO8czaJz) | **Medium** |
| 250 | **Array, Binary Search, Greedy** | [https://leetcode.com/problems/capacity-to-ship-packages-within-d-days](https://www.google.com/url?q=https://leetcode.com/problems/capacity-to-ship-packages-within-d-days&sa=D&source=editors&ust=1707482066899892&usg=AOvVaw0BBNe2h5I0Up9zGmJA8aqG) | **Medium** |
| 251 | **Two Pointers, String, String Matching** | [https://leetcode.com/problems/implement-strstr](https://www.google.com/url?q=https://leetcode.com/problems/implement-strstr&sa=D&source=editors&ust=1707482066900104&usg=AOvVaw0MbWwxI2ZPM5kCD_U2j014) | **Easy** |
| 252 | **Hash Table, String, Backtracking, Breadth-First Search** | [https://leetcode.com/problems/word-ladder-ii](https://www.google.com/url?q=https://leetcode.com/problems/word-ladder-ii&sa=D&source=editors&ust=1707482066900293&usg=AOvVaw0BAWp6aIuqMwB6w6cBB5Sg) | **Hard** |
| 253 | **Array, Sliding Window** | [https://leetcode.com/problems/subarray-product-less-than-k](https://www.google.com/url?q=https://leetcode.com/problems/subarray-product-less-than-k&sa=D&source=editors&ust=1707482066900500&usg=AOvVaw0zk0nv8BNvmfzcml4bVgpR) | **Medium** |
| 254 | **Array, Hash Table, Sorting** | [https://leetcode.com/problems/contains-duplicate](https://www.google.com/url?q=https://leetcode.com/problems/contains-duplicate&sa=D&source=editors&ust=1707482066900650&usg=AOvVaw3dz0SxxBnFxqOfi6SLR2kw) | **Easy** |
| 255 | **Linked List, Depth-First Search, Doubly-Linked List** | [https://leetcode.com/problems/flatten-a-multilevel-doubly-linked-list](https://www.google.com/url?q=https://leetcode.com/problems/flatten-a-multilevel-doubly-linked-list&sa=D&source=editors&ust=1707482066900856&usg=AOvVaw2NTGpRXkFV5A3s0-MSKUfs) | **Medium** |
| 256 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/populating-next-right-pointers-in-each-node-ii](https://www.google.com/url?q=https://leetcode.com/problems/populating-next-right-pointers-in-each-node-ii&sa=D&source=editors&ust=1707482066901024&usg=AOvVaw39oZUfZ9bRJ6zngrQgzXP8) | **Medium** |
| 257 | **Array, Divide and Conquer, Stack, Tree, Monotonic Stack, Binary Tree** | [https://leetcode.com/problems/maximum-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/maximum-binary-tree&sa=D&source=editors&ust=1707482066901215&usg=AOvVaw0NXvdZx0ZFsbTrwmf-ceG-) | **Medium** |
| 258 | **String, Stack** | [https://leetcode.com/problems/minimum-remove-to-make-valid-parentheses](https://www.google.com/url?q=https://leetcode.com/problems/minimum-remove-to-make-valid-parentheses&sa=D&source=editors&ust=1707482066901400&usg=AOvVaw3Kf2ouZK-0o8jxzQ4oMvdT) | **Medium** |
| 259 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/distribute-coins-in-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/distribute-coins-in-binary-tree&sa=D&source=editors&ust=1707482066901581&usg=AOvVaw17YSwm6UVIX0cE2fhx8VI-) | **Medium** |
| 260 | **Depth-First Search, Graph, Biconnected Component** | [https://leetcode.com/problems/critical-connections-in-a-network](https://www.google.com/url?q=https://leetcode.com/problems/critical-connections-in-a-network&sa=D&source=editors&ust=1707482066901780&usg=AOvVaw3f_AkbnnEQoVxkgXWdBiwH) | **Hard** |
| 261 | **Math, String, Stack, Recursion** | [https://leetcode.com/problems/basic-calculator](https://www.google.com/url?q=https://leetcode.com/problems/basic-calculator&sa=D&source=editors&ust=1707482066901916&usg=AOvVaw3iQtJ3Uo1Tg71FUs154JOp) | **Hard** |
| 262 | **Depth-First Search, Breadth-First Search, Union Find, Graph** | [https://leetcode.com/problems/redundant-connection](https://www.google.com/url?q=https://leetcode.com/problems/redundant-connection&sa=D&source=editors&ust=1707482066902345&usg=AOvVaw0_tjiQWNi9aK1dFXo0Gaed) | **Medium** |
| 263 | **Hash Table, String** | [https://leetcode.com/problems/jewels-and-stones](https://www.google.com/url?q=https://leetcode.com/problems/jewels-and-stones&sa=D&source=editors&ust=1707482066902632&usg=AOvVaw07MYEMRjhWCUitPx5c1y7F) | **Easy** |
| 264 | **Array, Dynamic Programming, Greedy, Sorting** | [https://leetcode.com/problems/non-overlapping-intervals](https://www.google.com/url?q=https://leetcode.com/problems/non-overlapping-intervals&sa=D&source=editors&ust=1707482066902836&usg=AOvVaw2boYTs9lWmg-j_nuyz_Hkj) | **Medium** |
| 265 | **Array, Depth-First Search, Breadth-First Search, Matrix** | [https://leetcode.com/problems/flood-fill](https://www.google.com/url?q=https://leetcode.com/problems/flood-fill&sa=D&source=editors&ust=1707482066903017&usg=AOvVaw3chbgtv47qdXBQHl_nrkrE) | **Easy** |
| 266 | **String** | [https://leetcode.com/problems/zigzag-conversion](https://www.google.com/url?q=https://leetcode.com/problems/zigzag-conversion&sa=D&source=editors&ust=1707482066903194&usg=AOvVaw19NeDG-IX0Xq5X-Svm6xrK) | **Medium** |
| 267 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/maximum-width-of-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/maximum-width-of-binary-tree&sa=D&source=editors&ust=1707482066903336&usg=AOvVaw2LToDadTyBVp0e1z4QHCkS) | **Medium** |
| 268 | **Math, Recursion** | [https://leetcode.com/problems/permutation-sequence](https://www.google.com/url?q=https://leetcode.com/problems/permutation-sequence&sa=D&source=editors&ust=1707482066903442&usg=AOvVaw2WYIB50d_-s-BAAPlXQkut) | **Hard** |
| 269 | **Array, Stack, Tree, Binary Search Tree, Monotonic Stack, Binary Tree** | [https://leetcode.com/problems/construct-binary-search-tree-from-preorder-traversal](https://www.google.com/url?q=https://leetcode.com/problems/construct-binary-search-tree-from-preorder-traversal&sa=D&source=editors&ust=1707482066903597&usg=AOvVaw2VBqHXcPbvRfAW1NfYXao9) | **Medium** |
| 270 | **Math, String, Dynamic Programming, Recursion, Memoization** | [https://leetcode.com/problems/different-ways-to-add-parentheses](https://www.google.com/url?q=https://leetcode.com/problems/different-ways-to-add-parentheses&sa=D&source=editors&ust=1707482066903735&usg=AOvVaw0xm44Jv0iiBGQ3bhyLOOHD) | **Medium** |
| 271 | **Array, Binary Search** | [https://leetcode.com/problems/search-in-rotated-sorted-array-ii](https://www.google.com/url?q=https://leetcode.com/problems/search-in-rotated-sorted-array-ii&sa=D&source=editors&ust=1707482066903842&usg=AOvVaw23uedxeyGtW68liHVvXOC-) | **Medium** |
| 272 | **Linked List, Math, Stack** | [https://leetcode.com/problems/add-two-numbers-ii](https://www.google.com/url?q=https://leetcode.com/problems/add-two-numbers-ii&sa=D&source=editors&ust=1707482066903959&usg=AOvVaw0BQ_AU4_OilA1eU9XSIZ_Z) | **Medium** |
| 273 | **Hash Table, Two Pointers, Tree, Depth-First Search, Breadth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/two-sum-iv-input-is-a-bst](https://www.google.com/url?q=https://leetcode.com/problems/two-sum-iv-input-is-a-bst&sa=D&source=editors&ust=1707482066904069&usg=AOvVaw2yfIt4z5vi25rlZBNWHP0L) | **Easy** |
| 274 | **Tree, Depth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/convert-bst-to-greater-tree](https://www.google.com/url?q=https://leetcode.com/problems/convert-bst-to-greater-tree&sa=D&source=editors&ust=1707482066904176&usg=AOvVaw189NulRGCJbm3R-l_QdE9m) | **Medium** |
| 275 | **Linked List, Two Pointers** | [https://leetcode.com/problems/partition-list](https://www.google.com/url?q=https://leetcode.com/problems/partition-list&sa=D&source=editors&ust=1707482066904284&usg=AOvVaw3OEsBBJJ4LqVata12KQ5F_) | **Medium** |
| 276 | **Math, Binary Search** | [https://leetcode.com/problems/sqrtx](https://www.google.com/url?q=https://leetcode.com/problems/sqrtx&sa=D&source=editors&ust=1707482066904398&usg=AOvVaw2UWj1gEk84GjKoBKNZL3yU) | **Easy** |
| 277 | **Hash Table, String** | [https://leetcode.com/problems/isomorphic-strings](https://www.google.com/url?q=https://leetcode.com/problems/isomorphic-strings&sa=D&source=editors&ust=1707482066904544&usg=AOvVaw28sMK5yJVI_82hyLKlSoL-) | **Easy** |
| 278 | **Array, Dynamic Programming, Binary Indexed Tree, Segment Tree** | [https://leetcode.com/problems/number-of-longest-increasing-subsequence](https://www.google.com/url?q=https://leetcode.com/problems/number-of-longest-increasing-subsequence&sa=D&source=editors&ust=1707482066904673&usg=AOvVaw3izYiO8QMfrWCGfQf49eiW) | **Medium** |
| 279 | **Array, Binary Search** | [https://leetcode.com/problems/binary-search](https://www.google.com/url?q=https://leetcode.com/problems/binary-search&sa=D&source=editors&ust=1707482066904799&usg=AOvVaw3K35fu_Qz8xsEaSVE8Cvsq) | **Easy** |
| 280 | **String, Dynamic Programming, Stack, Greedy** | [https://leetcode.com/problems/valid-parenthesis-string](https://www.google.com/url?q=https://leetcode.com/problems/valid-parenthesis-string&sa=D&source=editors&ust=1707482066904949&usg=AOvVaw39solAGuTK0j0_zAsBx3aK) | **Medium** |
| 281 | **String, String Matching** | [https://leetcode.com/problems/repeated-substring-pattern](https://www.google.com/url?q=https://leetcode.com/problems/repeated-substring-pattern&sa=D&source=editors&ust=1707482066905127&usg=AOvVaw09lOK0oLLPL-ZSrKHixb2K) | **Easy** |
| 282 | **Array, Two Pointers** | [https://leetcode.com/problems/remove-element](https://www.google.com/url?q=https://leetcode.com/problems/remove-element&sa=D&source=editors&ust=1707482066905243&usg=AOvVaw2HxyASVlp31N0XMmyzAr29) | **Easy** |
| 283 | **Array, Dynamic Programming** | [https://leetcode.com/problems/combination-sum-iv](https://www.google.com/url?q=https://leetcode.com/problems/combination-sum-iv&sa=D&source=editors&ust=1707482066905402&usg=AOvVaw1hkDwBI9zIA2R0sV6PSncU) | **Medium** |
| 284 | **String, Dynamic Programming** | [https://leetcode.com/problems/distinct-subsequences](https://www.google.com/url?q=https://leetcode.com/problems/distinct-subsequences&sa=D&source=editors&ust=1707482066905580&usg=AOvVaw1OlkqkjQOTI0B9Sbr8jARa) | **Hard** |
| 285 | **Array, Stack** | [https://leetcode.com/problems/asteroid-collision](https://www.google.com/url?q=https://leetcode.com/problems/asteroid-collision&sa=D&source=editors&ust=1707482066905760&usg=AOvVaw0E7Ia_jg5uGRttrxDSuZhO) | **Medium** |
| 286 | **Array, Math, Dynamic Programming, Sorting** | [https://leetcode.com/problems/largest-divisible-subset](https://www.google.com/url?q=https://leetcode.com/problems/largest-divisible-subset&sa=D&source=editors&ust=1707482066905910&usg=AOvVaw1V1GZqmUwObaLPQYxr5FSE) | **Medium** |
| 287 | **Array, Depth-First Search, Breadth-First Search, Matrix** | [https://leetcode.com/problems/pacific-atlantic-water-flow](https://www.google.com/url?q=https://leetcode.com/problems/pacific-atlantic-water-flow&sa=D&source=editors&ust=1707482066906034&usg=AOvVaw3UfCnOHIkA8FiKnMR8t7V_) | **Medium** |
| 288 | **Array, Binary Search, Stack, Monotonic Stack, Ordered Set** | [https://leetcode.com/problems/132-pattern](https://www.google.com/url?q=https://leetcode.com/problems/132-pattern&sa=D&source=editors&ust=1707482066906137&usg=AOvVaw1w5Swolghl6Z44CEATzffT) | **Medium** |
| 289 | **Array, Hash Table, Dynamic Programming** | [https://leetcode.com/problems/delete-and-earn](https://www.google.com/url?q=https://leetcode.com/problems/delete-and-earn&sa=D&source=editors&ust=1707482066906245&usg=AOvVaw26OeFxobJaGmjRATFZTbdf) | **Medium** |
| 290 | **Array, Dynamic Programming, Stack, Monotonic Stack** | [https://leetcode.com/problems/sum-of-subarray-minimums](https://www.google.com/url?q=https://leetcode.com/problems/sum-of-subarray-minimums&sa=D&source=editors&ust=1707482066906362&usg=AOvVaw0Ux0IZ_mPCm7ILMzqEBuqQ) | **Medium** |
| 291 | **Array, Dynamic Programming, Matrix** | [https://leetcode.com/problems/count-square-submatrices-with-all-ones](https://www.google.com/url?q=https://leetcode.com/problems/count-square-submatrices-with-all-ones&sa=D&source=editors&ust=1707482066906481&usg=AOvVaw1hQvyeko6D9iYnALUE1_Kk) | **Medium** |
| 292 | **String, Dynamic Programming** | [https://leetcode.com/problems/palindrome-partitioning-ii](https://www.google.com/url?q=https://leetcode.com/problems/palindrome-partitioning-ii&sa=D&source=editors&ust=1707482066906597&usg=AOvVaw3zbkp-C_KY7PzpAePAloPP) | **Hard** |
| 293 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/sum-of-left-leaves](https://www.google.com/url?q=https://leetcode.com/problems/sum-of-left-leaves&sa=D&source=editors&ust=1707482066906709&usg=AOvVaw3RltketOpvDfVLE01bBJuT) | **Easy** |
| 294 | **Bit Manipulation** | [https://leetcode.com/problems/hamming-distance](https://www.google.com/url?q=https://leetcode.com/problems/hamming-distance&sa=D&source=editors&ust=1707482066906811&usg=AOvVaw1-rJGSnxLVn2OEm9XwQHZl) | **Easy** |
| 295 | **Stack, Tree, Depth-First Search, Design, Queue, Iterator** | [https://leetcode.com/problems/flatten-nested-list-iterator](https://www.google.com/url?q=https://leetcode.com/problems/flatten-nested-list-iterator&sa=D&source=editors&ust=1707482066906917&usg=AOvVaw3TvJicaFpBdAR3B5Wfg0u-) | **Medium** |
| 296 | **String, Stack** | [https://leetcode.com/problems/score-of-parentheses](https://www.google.com/url?q=https://leetcode.com/problems/score-of-parentheses&sa=D&source=editors&ust=1707482066907024&usg=AOvVaw3VXCcXeCB9zgnrD6-l9_sI) | **Medium** |
| 297 | **Array, Hash Table, Two Pointers, String, Dynamic Programming** | [https://leetcode.com/problems/longest-string-chain](https://www.google.com/url?q=https://leetcode.com/problems/longest-string-chain&sa=D&source=editors&ust=1707482066907129&usg=AOvVaw3g-YyT0tdqVZMz6QZaavaF) | **Medium** |
| 298 | **Two Pointers, String** | [https://leetcode.com/problems/valid-palindrome](https://www.google.com/url?q=https://leetcode.com/problems/valid-palindrome&sa=D&source=editors&ust=1707482066907317&usg=AOvVaw10rTeFIzvsFzx8xj1fdhYq) | **Easy** |
| 299 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/longest-univalue-path](https://www.google.com/url?q=https://leetcode.com/problems/longest-univalue-path&sa=D&source=editors&ust=1707482066907482&usg=AOvVaw3hSXjPPyDcF4aZObiL_3x1) | **Medium** |
| 300 | **String, Backtracking, Bit Manipulation** | [https://leetcode.com/problems/letter-case-permutation](https://www.google.com/url?q=https://leetcode.com/problems/letter-case-permutation&sa=D&source=editors&ust=1707482066907590&usg=AOvVaw2tebB0LJeYvuOf_43rTu7j) | **Medium** |
| 301 | **Tree, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-level-order-traversal-ii](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-level-order-traversal-ii&sa=D&source=editors&ust=1707482066907745&usg=AOvVaw0J-ZZr49Spko7Xu0OTl7Lh) | **Medium** |
| 302 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/find-duplicate-subtrees](https://www.google.com/url?q=https://leetcode.com/problems/find-duplicate-subtrees&sa=D&source=editors&ust=1707482066907914&usg=AOvVaw0drQkHHObl3JDvhlymRRDV) | **Medium** |
| 303 | **Dynamic Programming, Stack, Greedy, Monotonic Stack** | [https://leetcode.com/problems/minimum-cost-tree-from-leaf-values](https://www.google.com/url?q=https://leetcode.com/problems/minimum-cost-tree-from-leaf-values&sa=D&source=editors&ust=1707482066908092&usg=AOvVaw15ZT88IhyRF8LGHR5KR7Ux) | **Medium** |
| 304 | **Hash Table, Linked List, Design, Doubly-Linked List** | [https://leetcode.com/problems/lfu-cache](https://www.google.com/url?q=https://leetcode.com/problems/lfu-cache&sa=D&source=editors&ust=1707482066908271&usg=AOvVaw2E642GDn0r33oBvS7boYse) | **Hard** |
| 305 | **Array, Binary Search, Dynamic Programming, Sorting** | [https://leetcode.com/problems/russian-doll-envelopes](https://www.google.com/url?q=https://leetcode.com/problems/russian-doll-envelopes&sa=D&source=editors&ust=1707482066908451&usg=AOvVaw2bCQ-wYbFc3xUtLwUnSU1x) | **Hard** |
| 306 | **Array, Binary Search, Dynamic Programming, Sorting** | [https://leetcode.com/problems/maximum-profit-in-job-scheduling](https://www.google.com/url?q=https://leetcode.com/problems/maximum-profit-in-job-scheduling&sa=D&source=editors&ust=1707482066908620&usg=AOvVaw0rPy0-FGQ9PWKyiudXKVvc) | **Hard** |
| 307 | **Math, String, Simulation** | [https://leetcode.com/problems/add-strings](https://www.google.com/url?q=https://leetcode.com/problems/add-strings&sa=D&source=editors&ust=1707482066908788&usg=AOvVaw0RgpVyB540zbt9oIQxQawM) | **Easy** |
| 308 | **Array, Hash Table, Counting** | [https://leetcode.com/problems/pairs-of-songs-with-total-durations-divisible-by-60](https://www.google.com/url?q=https://leetcode.com/problems/pairs-of-songs-with-total-durations-divisible-by-60&sa=D&source=editors&ust=1707482066908972&usg=AOvVaw3lFDUP724NaA8PJ7WSFbNW) | **Medium** |
| 309 | **Stack, Design, Queue** | [https://leetcode.com/problems/implement-queue-using-stacks](https://www.google.com/url?q=https://leetcode.com/problems/implement-queue-using-stacks&sa=D&source=editors&ust=1707482066909150&usg=AOvVaw18v0fnysxDJPajVEl_2dTd) | **Easy** |
| 310 | **Array, Hash Table, Bit Manipulation, Trie** | [https://leetcode.com/problems/maximum-xor-of-two-numbers-in-an-array](https://www.google.com/url?q=https://leetcode.com/problems/maximum-xor-of-two-numbers-in-an-array&sa=D&source=editors&ust=1707482066909328&usg=AOvVaw26cOhAKStV1NmHI0c-CUr2) | **Medium** |
| 311 | **Hash Table, Math, String** | [https://leetcode.com/problems/roman-to-integer](https://www.google.com/url?q=https://leetcode.com/problems/roman-to-integer&sa=D&source=editors&ust=1707482066909502&usg=AOvVaw0f8NF6K9qzXOmzfoe2VCn4) | **Easy** |
| 312 | **Array, Hash Table, Graph** | [https://leetcode.com/problems/find-the-town-judge](https://www.google.com/url?q=https://leetcode.com/problems/find-the-town-judge&sa=D&source=editors&ust=1707482066909664&usg=AOvVaw3y2S903TAZWUiP-YIos_QU) | **Easy** |
| 313 | **Array, Heap (Priority Queue)** | [https://leetcode.com/problems/find-k-pairs-with-smallest-sums](https://www.google.com/url?q=https://leetcode.com/problems/find-k-pairs-with-smallest-sums&sa=D&source=editors&ust=1707482066909835&usg=AOvVaw0wGmW6ewvTOz6_6KDmwAHi) | **Medium** |
| 314 | **Array, Two Pointers, Sorting** | [https://leetcode.com/problems/merge-sorted-array](https://www.google.com/url?q=https://leetcode.com/problems/merge-sorted-array&sa=D&source=editors&ust=1707482066910013&usg=AOvVaw2aM0wImloqp8K_1GaF90DS) | **Easy** |
| 315 | **String, Stack** | [https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string](https://www.google.com/url?q=https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string&sa=D&source=editors&ust=1707482066910182&usg=AOvVaw1GvSHvo02w6gs3Aawh0zo8) | **Easy** |
| 316 | **Two Pointers, String** | [https://leetcode.com/problems/reverse-words-in-a-string](https://www.google.com/url?q=https://leetcode.com/problems/reverse-words-in-a-string&sa=D&source=editors&ust=1707482066910352&usg=AOvVaw0_9-VTbGSSS9B8LrJy7XzD) | **Medium** |
| 317 | **Array, Hash Table** | [https://leetcode.com/problems/4sum-ii](https://www.google.com/url?q=https://leetcode.com/problems/4sum-ii&sa=D&source=editors&ust=1707482066910521&usg=AOvVaw1DvKcoxAGLc1gT8AszzAOs) | **Medium** |
| 318 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/average-of-levels-in-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/average-of-levels-in-binary-tree&sa=D&source=editors&ust=1707482066910692&usg=AOvVaw3AK9r64NLKHpSoa0MHU1hn) | **Easy** |
| 319 | **Array, Prefix Sum** | [https://leetcode.com/problems/find-pivot-index](https://www.google.com/url?q=https://leetcode.com/problems/find-pivot-index&sa=D&source=editors&ust=1707482066910866&usg=AOvVaw0hLt3HIRk7-dCF4-hf-clC) | **Easy** |
| 320 | **Array, Hash Table, Sorting, Counting** | [https://leetcode.com/problems/how-many-numbers-are-smaller-than-the-current-number](https://www.google.com/url?q=https://leetcode.com/problems/how-many-numbers-are-smaller-than-the-current-number&sa=D&source=editors&ust=1707482066911040&usg=AOvVaw3uM9VBUeTZa3AI0zhy299k) | **Easy** |
| 321 | **Hash Table, Math, String** | [https://leetcode.com/problems/integer-to-roman](https://www.google.com/url?q=https://leetcode.com/problems/integer-to-roman&sa=D&source=editors&ust=1707482066911212&usg=AOvVaw1rt15XNp76fGCaJ2z6mOGI) | **Medium** |
| 322 | **Depth-First Search, Breadth-First Search, Graph** | [https://leetcode.com/problems/keys-and-rooms](https://www.google.com/url?q=https://leetcode.com/problems/keys-and-rooms&sa=D&source=editors&ust=1707482066911392&usg=AOvVaw3X9kVENTudVdBjRV9TT1Zu) | **Medium** |
| 323 | **Math, Bit Manipulation, Recursion** | [https://leetcode.com/problems/power-of-two](https://www.google.com/url?q=https://leetcode.com/problems/power-of-two&sa=D&source=editors&ust=1707482066911576&usg=AOvVaw2QHjXa-yLVgoxJ2bHwOM3N) | **Easy** |
| 324 | **Array, Hash Table, Sliding Window, Counting** | [https://leetcode.com/problems/subarrays-with-k-different-integers](https://www.google.com/url?q=https://leetcode.com/problems/subarrays-with-k-different-integers&sa=D&source=editors&ust=1707482066911744&usg=AOvVaw0TbV6NafoV7y4wbsZ01ExP) | **Hard** |
| 325 | **Array, Math, Dynamic Programming, Recursion, Game Theory** | [https://leetcode.com/problems/predict-the-winner](https://www.google.com/url?q=https://leetcode.com/problems/predict-the-winner&sa=D&source=editors&ust=1707482066911922&usg=AOvVaw2q1kTZ3_JS1Jl7HlWOLbFZ) | **Medium** |
| 326 | **Array, Divide and Conquer, Dynamic Programming, Queue, Monotonic Queue** | [https://leetcode.com/problems/maximum-sum-circular-subarray](https://www.google.com/url?q=https://leetcode.com/problems/maximum-sum-circular-subarray&sa=D&source=editors&ust=1707482066912104&usg=AOvVaw2m63SgNU2pR3hQNz9GPyjW) | **Medium** |
| 327 | **Math, Dynamic Programming, Recursion, Memoization** | [https://leetcode.com/problems/fibonacci-number](https://www.google.com/url?q=https://leetcode.com/problems/fibonacci-number&sa=D&source=editors&ust=1707482066912330&usg=AOvVaw2hZhsVb_wXXU7-cJ0_9hGT) | **Easy** |
| 328 | **Divide and Conquer, Bit Manipulation** | [https://leetcode.com/problems/reverse-bits](https://www.google.com/url?q=https://leetcode.com/problems/reverse-bits&sa=D&source=editors&ust=1707482066912504&usg=AOvVaw07oBwFUM8_lTjJ2O02ty8u) | **Easy** |
| 329 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/delete-nodes-and-return-forest](https://www.google.com/url?q=https://leetcode.com/problems/delete-nodes-and-return-forest&sa=D&source=editors&ust=1707482066912676&usg=AOvVaw3xeuhPghsRgNzBUxKWFvg6) | **Medium** |
| 330 | **Array, Binary Search, Queue, Sliding Window, Heap (Priority Queue), Prefix Sum, Monotonic Queue** | [https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k](https://www.google.com/url?q=https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k&sa=D&source=editors&ust=1707482066912850&usg=AOvVaw30CjMUx1JrywK13Jq-4PZY) | **Hard** |
| 331 | **Array, Binary Search** | [https://leetcode.com/problems/find-minimum-in-rotated-sorted-array-ii](https://www.google.com/url?q=https://leetcode.com/problems/find-minimum-in-rotated-sorted-array-ii&sa=D&source=editors&ust=1707482066913026&usg=AOvVaw3WQ0d1JxUHavbC7_sAdZDe) | **Hard** |
| 332 | **Hash Table, String** | [https://leetcode.com/problems/word-pattern](https://www.google.com/url?q=https://leetcode.com/problems/word-pattern&sa=D&source=editors&ust=1707482066913204&usg=AOvVaw0WV7qMPKwr1e7iD11UPnN4) | **Easy** |
| 333 | **Array, Dynamic Programming** | [https://leetcode.com/problems/arithmetic-slices](https://www.google.com/url?q=https://leetcode.com/problems/arithmetic-slices&sa=D&source=editors&ust=1707482066913379&usg=AOvVaw2SnnaQGrFPHUhZioY6Ob2g) | **Medium** |
| 334 | **Array, String, Dynamic Programming** | [https://leetcode.com/problems/ones-and-zeroes](https://www.google.com/url?q=https://leetcode.com/problems/ones-and-zeroes&sa=D&source=editors&ust=1707482066913546&usg=AOvVaw0m2Mya8D6Qer99EK2MbDXn) | **Medium** |
| 335 | **Array, Backtracking, Bit Manipulation, Matrix** | [https://leetcode.com/problems/unique-paths-iii](https://www.google.com/url?q=https://leetcode.com/problems/unique-paths-iii&sa=D&source=editors&ust=1707482066913718&usg=AOvVaw27t12zCoiA52Wk0nmptYIe) | **Hard** |
| 336 | **Array, Design, Binary Indexed Tree, Segment Tree** | [https://leetcode.com/problems/range-sum-query-mutable](https://www.google.com/url?q=https://leetcode.com/problems/range-sum-query-mutable&sa=D&source=editors&ust=1707482066913891&usg=AOvVaw1nMdTxIwTD6ytVqc8xE49r) | **Medium** |
| 337 | **Hash Table, Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/vertical-order-traversal-of-a-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/vertical-order-traversal-of-a-binary-tree&sa=D&source=editors&ust=1707482066914064&usg=AOvVaw07iRHz2BiftP8azTqALe7h) | **Hard** |
| 338 | **Array, Greedy** | [https://leetcode.com/problems/candy](https://www.google.com/url?q=https://leetcode.com/problems/candy&sa=D&source=editors&ust=1707482066914232&usg=AOvVaw30FEQba_a3dwbZGYUTE9G8) | **Hard** |
| 339 | **Array, Two Pointers** | [https://leetcode.com/problems/remove-duplicates-from-sorted-array-ii](https://www.google.com/url?q=https://leetcode.com/problems/remove-duplicates-from-sorted-array-ii&sa=D&source=editors&ust=1707482066914410&usg=AOvVaw25FOEAL7RQ7bByTKBfKf1W) | **Medium** |
| 340 | **Hash Table, String, Trie, Sorting** | [https://leetcode.com/problems/number-of-matching-subsequences](https://www.google.com/url?q=https://leetcode.com/problems/number-of-matching-subsequences&sa=D&source=editors&ust=1707482066914585&usg=AOvVaw399Cmp8D-9DcZLr16ZW5q4) | **Medium** |
| 341 | **Array, Binary Search** | [https://leetcode.com/problems/koko-eating-bananas](https://www.google.com/url?q=https://leetcode.com/problems/koko-eating-bananas&sa=D&source=editors&ust=1707482066914767&usg=AOvVaw03I2JZDc4nXH5ElA0071OP) | **Medium** |
| 342 | **Array, Hash Table, String, Breadth-First Search** | [https://leetcode.com/problems/open-the-lock](https://www.google.com/url?q=https://leetcode.com/problems/open-the-lock&sa=D&source=editors&ust=1707482066914939&usg=AOvVaw2PMcGvXyWGywPQWWg1w9bV) | **Medium** |
| 343 | **String, Tree, Depth-First Search, Breadth-First Search, Design, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/serialize-and-deserialize-bst](https://www.google.com/url?q=https://leetcode.com/problems/serialize-and-deserialize-bst&sa=D&source=editors&ust=1707482066915107&usg=AOvVaw1sYS3n5shzK5CrB2bApx9b) | **Medium** |
| 344 | **Array, Backtracking** | [https://leetcode.com/problems/combination-sum-iii](https://www.google.com/url?q=https://leetcode.com/problems/combination-sum-iii&sa=D&source=editors&ust=1707482066915280&usg=AOvVaw192ZNVvVidcJU2MSTWGGLm) | **Medium** |
| 345 | **Array, Hash Table, String, Trie** | [https://leetcode.com/problems/palindrome-pairs](https://www.google.com/url?q=https://leetcode.com/problems/palindrome-pairs&sa=D&source=editors&ust=1707482066915462&usg=AOvVaw1a9beA4JTHK2nTJ_fxnTnN) | **Hard** |
| 346 | **Array, Hash Table, String** | [https://leetcode.com/problems/verifying-an-alien-dictionary](https://www.google.com/url?q=https://leetcode.com/problems/verifying-an-alien-dictionary&sa=D&source=editors&ust=1707482066915633&usg=AOvVaw0jzI2hrU2l0XJ1A7n63xFh) | **Easy** |
| 347 | **Array, Math, Stack** | [https://leetcode.com/problems/evaluate-reverse-polish-notation](https://www.google.com/url?q=https://leetcode.com/problems/evaluate-reverse-polish-notation&sa=D&source=editors&ust=1707482066915800&usg=AOvVaw2KK2ul3Ejrzck606r_CnrX) | **Medium** |
| 348 | **Array, Breadth-First Search, Heap (Priority Queue), Matrix** | [https://leetcode.com/problems/trapping-rain-water-ii](https://www.google.com/url?q=https://leetcode.com/problems/trapping-rain-water-ii&sa=D&source=editors&ust=1707482066915981&usg=AOvVaw099_rnM_jnBrobFX_AWrEO) | **Hard** |
| 349 | **Array, Sliding Window, Prefix Sum** | [https://leetcode.com/problems/maximum-points-you-can-obtain-from-cards](https://www.google.com/url?q=https://leetcode.com/problems/maximum-points-you-can-obtain-from-cards&sa=D&source=editors&ust=1707482066916158&usg=AOvVaw1N5FCJI_2LPCWIyExC4yAL) | **Medium** |
| 350 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/cousins-in-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/cousins-in-binary-tree&sa=D&source=editors&ust=1707482066916358&usg=AOvVaw25vzLPODpyA8ahJrHmKmbL) | **Easy** |
| 351 | **Math, Bit Manipulation** | [https://leetcode.com/problems/divide-two-integers](https://www.google.com/url?q=https://leetcode.com/problems/divide-two-integers&sa=D&source=editors&ust=1707482066916529&usg=AOvVaw3Yxy8KeOgwm5exZ6wlHfNi) | **Medium** |
| 352 | **String, Dynamic Programming** | [https://leetcode.com/problems/delete-operation-for-two-strings](https://www.google.com/url?q=https://leetcode.com/problems/delete-operation-for-two-strings&sa=D&source=editors&ust=1707482066916699&usg=AOvVaw1CZTbHKVF9ks4ziK02oArI) | **Medium** |
| 353 | **Array, Greedy, Sorting** | [https://leetcode.com/problems/minimum-number-of-arrows-to-burst-balloons](https://www.google.com/url?q=https://leetcode.com/problems/minimum-number-of-arrows-to-burst-balloons&sa=D&source=editors&ust=1707482066916882&usg=AOvVaw2luI4oaR36GLXZDkhjCuzP) | **Medium** |
| 354 | **Array, Hash Table, Prefix Sum** | [https://leetcode.com/problems/subarray-sums-divisible-by-k](https://www.google.com/url?q=https://leetcode.com/problems/subarray-sums-divisible-by-k&sa=D&source=editors&ust=1707482066917065&usg=AOvVaw2AA1m-LhZYmt0Yr16zF1O_) | **Medium** |
| 355 | **Two Pointers, String** | [https://leetcode.com/problems/count-binary-substrings](https://www.google.com/url?q=https://leetcode.com/problems/count-binary-substrings&sa=D&source=editors&ust=1707482066917237&usg=AOvVaw2wxlPP5XGrgPmmiZszuS5Q) | **Easy** |
| 356 | **String, Backtracking** | [https://leetcode.com/problems/restore-ip-addresses](https://www.google.com/url?q=https://leetcode.com/problems/restore-ip-addresses&sa=D&source=editors&ust=1707482066917409&usg=AOvVaw3buxI1vr7PO1VMtL4098W0) | **Medium** |
| 357 | **Array, Matrix, Simulation** | [https://leetcode.com/problems/spiral-matrix-ii](https://www.google.com/url?q=https://leetcode.com/problems/spiral-matrix-ii&sa=D&source=editors&ust=1707482066917585&usg=AOvVaw1NJeB3HOn20i4-3Xm1DHLb) | **Medium** |
| 358 | **Hash Table, Stack, Design, Ordered Set** | [https://leetcode.com/problems/maximum-frequency-stack](https://www.google.com/url?q=https://leetcode.com/problems/maximum-frequency-stack&sa=D&source=editors&ust=1707482066917765&usg=AOvVaw2xmxVEfv9xvA_Nt0ln_nqU) | **Hard** |
| 359 | **Math, Dynamic Programming** | [https://leetcode.com/problems/integer-break](https://www.google.com/url?q=https://leetcode.com/problems/integer-break&sa=D&source=editors&ust=1707482066917946&usg=AOvVaw1C8ga2WJd3dwlAkruEvNcA) | **Medium** |
| 360 | **Array, Two Pointers, Sorting** | [https://leetcode.com/problems/sort-array-by-parity](https://www.google.com/url?q=https://leetcode.com/problems/sort-array-by-parity&sa=D&source=editors&ust=1707482066918113&usg=AOvVaw08fQjG446ngFBpocx3Qato) | **Easy** |
| 361 | **Dynamic Programming, Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-cameras](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-cameras&sa=D&source=editors&ust=1707482066918281&usg=AOvVaw1BAtR9ezXIJvQaDSxSny67) | **Hard** |
| 362 | **Array, Dynamic Programming, Greedy** | [https://leetcode.com/problems/wiggle-subsequence](https://www.google.com/url?q=https://leetcode.com/problems/wiggle-subsequence&sa=D&source=editors&ust=1707482066918458&usg=AOvVaw2yInH-iKshq-RlIpNffgt6) | **Medium** |
| 363 | **Array, Depth-First Search, Breadth-First Search** | [https://leetcode.com/problems/jump-game-iii](https://www.google.com/url?q=https://leetcode.com/problems/jump-game-iii&sa=D&source=editors&ust=1707482066918625&usg=AOvVaw1HaIcbr4RGoiDPr0peAkLN) | **Medium** |
| 364 | **Bit Manipulation** | [https://leetcode.com/problems/number-of-1-bits](https://www.google.com/url?q=https://leetcode.com/problems/number-of-1-bits&sa=D&source=editors&ust=1707482066918801&usg=AOvVaw0XDwlk8DRI5BVNUJIHXtnu) | **Easy** |
| 365 | **Tree, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/insert-into-a-binary-search-tree](https://www.google.com/url?q=https://leetcode.com/problems/insert-into-a-binary-search-tree&sa=D&source=editors&ust=1707482066918973&usg=AOvVaw0t8riSWMvaUa2f6DAwMcZ_) | **Medium** |
| 366 | **Array, Math, Sorting** | [https://leetcode.com/problems/maximum-product-of-three-numbers](https://www.google.com/url?q=https://leetcode.com/problems/maximum-product-of-three-numbers&sa=D&source=editors&ust=1707482066919155&usg=AOvVaw0c4m3cCy7RUejJb10Ni-Ro) | **Easy** |
| 367 | **Math, String, Backtracking** | [https://leetcode.com/problems/expression-add-operators](https://www.google.com/url?q=https://leetcode.com/problems/expression-add-operators&sa=D&source=editors&ust=1707482066919326&usg=AOvVaw1c3k28ebEVBhbwVNeX50Wi) | **Hard** |
| 368 | **Math, String** | [https://leetcode.com/problems/excel-sheet-column-title](https://www.google.com/url?q=https://leetcode.com/problems/excel-sheet-column-title&sa=D&source=editors&ust=1707482066919500&usg=AOvVaw0XQNEUvFxw8bgHww__uu-T) | **Easy** |
| 369 | **Array, Binary Search, Divide and Conquer, Binary Indexed Tree, Segment Tree, Merge Sort, Ordered Set** | [https://leetcode.com/problems/reverse-pairs](https://www.google.com/url?q=https://leetcode.com/problems/reverse-pairs&sa=D&source=editors&ust=1707482066919685&usg=AOvVaw0-OGuf4LSB8Hg00HiqzWxu) | **Hard** |
| 370 | **Dynamic Programming, Tree, Depth-First Search, Graph** | [https://leetcode.com/problems/sum-of-distances-in-tree](https://www.google.com/url?q=https://leetcode.com/problems/sum-of-distances-in-tree&sa=D&source=editors&ust=1707482066919864&usg=AOvVaw23oWUUcjVnetk3nDNLyExj) | **Hard** |
| 371 | **Array, Greedy, Sorting** | [https://leetcode.com/problems/two-city-scheduling](https://www.google.com/url?q=https://leetcode.com/problems/two-city-scheduling&sa=D&source=editors&ust=1707482066920035&usg=AOvVaw08F9PTczoLBkxI57oE3bKJ) | **Medium** |
| 372 | **String, Stack** | [https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string-ii](https://www.google.com/url?q=https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string-ii&sa=D&source=editors&ust=1707482066920203&usg=AOvVaw0wgI3u6v2rzdPA4-xb4wyW) | **Medium** |
| 373 | **Array, Design, Matrix, Prefix Sum** | [https://leetcode.com/problems/range-sum-query-2d-immutable](https://www.google.com/url?q=https://leetcode.com/problems/range-sum-query-2d-immutable&sa=D&source=editors&ust=1707482066920387&usg=AOvVaw25Izgr06kOTUhPGFMfsDa3) | **Medium** |
| 374 | **Math, Dynamic Programming** | [https://leetcode.com/problems/2-keys-keyboard](https://www.google.com/url?q=https://leetcode.com/problems/2-keys-keyboard&sa=D&source=editors&ust=1707482066920561&usg=AOvVaw1t8GAlInrhTsv6kT9P-Fr3) | **Medium** |
| 375 | **Array, Hash Table, Two Pointers, Binary Search, Sorting** | [https://leetcode.com/problems/intersection-of-two-arrays](https://www.google.com/url?q=https://leetcode.com/problems/intersection-of-two-arrays&sa=D&source=editors&ust=1707482066920727&usg=AOvVaw2zsJnvwkEMeqpxb502fQWE) | **Easy** |
| 376 | **Array, Stack, Simulation** | [https://leetcode.com/problems/validate-stack-sequences](https://www.google.com/url?q=https://leetcode.com/problems/validate-stack-sequences&sa=D&source=editors&ust=1707482066920896&usg=AOvVaw10UkFIYE3z5Tuz_wjqCTIz) | **Medium** |
| 377 | **Hash Table, String, Greedy** | [https://leetcode.com/problems/longest-palindrome](https://www.google.com/url?q=https://leetcode.com/problems/longest-palindrome&sa=D&source=editors&ust=1707482066921068&usg=AOvVaw2SFob4mB9DeGdwRvlweNUc) | **Easy** |
| 378 | **Tree, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/search-in-a-binary-search-tree](https://www.google.com/url?q=https://leetcode.com/problems/search-in-a-binary-search-tree&sa=D&source=editors&ust=1707482066921240&usg=AOvVaw0tMsS8Vq6Lsw3fz6Efwyxz) | **Easy** |
| 379 | **Array, Dynamic Programming, Greedy, Heap (Priority Queue)** | [https://leetcode.com/problems/minimum-number-of-refueling-stops](https://www.google.com/url?q=https://leetcode.com/problems/minimum-number-of-refueling-stops&sa=D&source=editors&ust=1707482066921439&usg=AOvVaw2QEONGg8clEw572T77Y2e-) | **Hard** |
| 380 | **Array, Two Pointers, Binary Search, Greedy, Sorting** | [https://leetcode.com/problems/valid-triangle-number](https://www.google.com/url?q=https://leetcode.com/problems/valid-triangle-number&sa=D&source=editors&ust=1707482066921614&usg=AOvVaw2Xf4znA2629uuXnULYzhC8) | **Medium** |
| 381 | **Dynamic Programming, Tree, Recursion, Memoization, Binary Tree** | [https://leetcode.com/problems/all-possible-full-binary-trees](https://www.google.com/url?q=https://leetcode.com/problems/all-possible-full-binary-trees&sa=D&source=editors&ust=1707482066921787&usg=AOvVaw3EqimoK3I0KXdM2lt2xFzO) | **Medium** |
| 382 | **Two Pointers, String** | [https://leetcode.com/problems/reverse-words-in-a-string-iii](https://www.google.com/url?q=https://leetcode.com/problems/reverse-words-in-a-string-iii&sa=D&source=editors&ust=1707482066921955&usg=AOvVaw05InAUTiTiRxwX7hy6j4vB) | **Easy** |
| 383 | **Math, String** | [https://leetcode.com/problems/excel-sheet-column-number](https://www.google.com/url?q=https://leetcode.com/problems/excel-sheet-column-number&sa=D&source=editors&ust=1707482066922118&usg=AOvVaw0veL-EiGVLPjynKGRzIfHR) | **Easy** |
| 384 | **Array, Dynamic Programming, Matrix** | [https://leetcode.com/problems/cherry-pickup](https://www.google.com/url?q=https://leetcode.com/problems/cherry-pickup&sa=D&source=editors&ust=1707482066922269&usg=AOvVaw1-XvL6F--jLtom4fbqq1zl) | **Hard** |
| 385 | **Array, Hash Table, Linked List, Design, Hash Function** | [https://leetcode.com/problems/design-hashmap](https://www.google.com/url?q=https://leetcode.com/problems/design-hashmap&sa=D&source=editors&ust=1707482066922378&usg=AOvVaw2qLUPfUija0-byJW-q2E3n) | **Easy** |
| 386 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/binary-tree-pruning](https://www.google.com/url?q=https://leetcode.com/problems/binary-tree-pruning&sa=D&source=editors&ust=1707482066922479&usg=AOvVaw07s4bu52Ch_wQiLe6YeBzm) | **Medium** |
| 387 | **Math, Bit Manipulation** | [https://leetcode.com/problems/sum-of-two-integers](https://www.google.com/url?q=https://leetcode.com/problems/sum-of-two-integers&sa=D&source=editors&ust=1707482066922712&usg=AOvVaw0rrsgEfiXWrM0zXHYdbk6E) | **Medium** |
| 388 | **Stack, Design, Monotonic Stack, Data Stream** | [https://leetcode.com/problems/online-stock-span](https://www.google.com/url?q=https://leetcode.com/problems/online-stock-span&sa=D&source=editors&ust=1707482066922814&usg=AOvVaw1gKtIg8b8iAfzif0gzl8sD) | **Medium** |
| 389 | **Linked List, Math** | [https://leetcode.com/problems/convert-binary-number-in-a-linked-list-to-integer](https://www.google.com/url?q=https://leetcode.com/problems/convert-binary-number-in-a-linked-list-to-integer&sa=D&source=editors&ust=1707482066922917&usg=AOvVaw0w6QfRQXHh817y3betHL1p) | **Easy** |
| 390 | **Tree, Depth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/binary-search-tree-to-greater-sum-tree](https://www.google.com/url?q=https://leetcode.com/problems/binary-search-tree-to-greater-sum-tree&sa=D&source=editors&ust=1707482066923029&usg=AOvVaw3QG8ESIWYJh7QJYGZZXDWj) | **Medium** |
| 391 | **Array** | [https://leetcode.com/problems/max-consecutive-ones](https://www.google.com/url?q=https://leetcode.com/problems/max-consecutive-ones&sa=D&source=editors&ust=1707482066923147&usg=AOvVaw3fyRXM5oJY9pSQrpBc5Z6k) | **Easy** |
| 392 | **Array, Prefix Sum** | [https://leetcode.com/problems/running-sum-of-1d-array](https://www.google.com/url?q=https://leetcode.com/problems/running-sum-of-1d-array&sa=D&source=editors&ust=1707482066923266&usg=AOvVaw0yTdLjm0-H_XPo9SO_yQxa) | **Easy** |
| 393 | **Math, Greedy** | [https://leetcode.com/problems/maximum-swap](https://www.google.com/url?q=https://leetcode.com/problems/maximum-swap&sa=D&source=editors&ust=1707482066923427&usg=AOvVaw2ZAm6BImXp4oVMfJUDveD8) | **Medium** |
| 394 | **String, Rolling Hash, String Matching, Hash Function** | [https://leetcode.com/problems/shortest-palindrome](https://www.google.com/url?q=https://leetcode.com/problems/shortest-palindrome&sa=D&source=editors&ust=1707482066923566&usg=AOvVaw1JIHfSmcCgoYn3PZQxf4Aa) | **Hard** |
| 395 | **Depth-First Search, Union Find, Graph** | [https://leetcode.com/problems/most-stones-removed-with-same-row-or-column](https://www.google.com/url?q=https://leetcode.com/problems/most-stones-removed-with-same-row-or-column&sa=D&source=editors&ust=1707482066923682&usg=AOvVaw36TLIpnmHByW6BvNisoDGL) | **Medium** |
| 396 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/maximum-difference-between-node-and-ancestor](https://www.google.com/url?q=https://leetcode.com/problems/maximum-difference-between-node-and-ancestor&sa=D&source=editors&ust=1707482066923795&usg=AOvVaw1vK1joxCVbO-ELHR5CH22-) | **Medium** |
| 397 | **Array, Dynamic Programming** | [https://leetcode.com/problems/frog-jump](https://www.google.com/url?q=https://leetcode.com/problems/frog-jump&sa=D&source=editors&ust=1707482066923905&usg=AOvVaw2xGdqoOUGElMpOXxLA1dui) | **Hard** |
| 398 | **Array, Dynamic Programming** | [https://leetcode.com/problems/pascals-triangle-ii](https://www.google.com/url?q=https://leetcode.com/problems/pascals-triangle-ii&sa=D&source=editors&ust=1707482066924027&usg=AOvVaw3PbLKDp89m4haF2OhnE--O) | **Easy** |
| 399 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/deepest-leaves-sum](https://www.google.com/url?q=https://leetcode.com/problems/deepest-leaves-sum&sa=D&source=editors&ust=1707482066924195&usg=AOvVaw2UDCxAdd2jhCP8SlMN5zcG) | **Medium** |
| 400 | **Array, Depth-First Search, Breadth-First Search, Matrix** | [https://leetcode.com/problems/shortest-bridge](https://www.google.com/url?q=https://leetcode.com/problems/shortest-bridge&sa=D&source=editors&ust=1707482066924372&usg=AOvVaw3yESk8rLpmW2MXi8JWvA4f) | **Medium** |
| 401 | **Bit Manipulation** | [https://leetcode.com/problems/bitwise-and-of-numbers-range](https://www.google.com/url?q=https://leetcode.com/problems/bitwise-and-of-numbers-range&sa=D&source=editors&ust=1707482066924543&usg=AOvVaw109FUSwLNx7xg7zTnB1Q-y) | **Medium** |
| 402 | **Math, Binary Search, Dynamic Programming** | [https://leetcode.com/problems/super-egg-drop](https://www.google.com/url?q=https://leetcode.com/problems/super-egg-drop&sa=D&source=editors&ust=1707482066924786&usg=AOvVaw3lfZuGIU841YI_s882MhYY) | **Hard** |
| 403 | **Array, Sliding Window, Sorting, Bucket Sort, Ordered Set** | [https://leetcode.com/problems/contains-duplicate-iii](https://www.google.com/url?q=https://leetcode.com/problems/contains-duplicate-iii&sa=D&source=editors&ust=1707482066924910&usg=AOvVaw2GKSnrz3eGnSQtxGblb999) | **Medium** |
| 404 | **Math, String, Simulation** | [https://leetcode.com/problems/robot-bounded-in-circle](https://www.google.com/url?q=https://leetcode.com/problems/robot-bounded-in-circle&sa=D&source=editors&ust=1707482066925028&usg=AOvVaw2Dq1l2Wsuy3003BZah-uMu) | **Medium** |
| 405 | **Stack, Tree, Depth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/increasing-order-search-tree](https://www.google.com/url?q=https://leetcode.com/problems/increasing-order-search-tree&sa=D&source=editors&ust=1707482066925133&usg=AOvVaw17DkB92_CkjhGWfhsTl6q_) | **Easy** |
| 406 | **Array, Hash Table, String** | [https://leetcode.com/problems/find-common-characters](https://www.google.com/url?q=https://leetcode.com/problems/find-common-characters&sa=D&source=editors&ust=1707482066925244&usg=AOvVaw1ENsi7pvJI8_df72DQFGhH) | **Easy** |
| 407 | **Array, Two Pointers, String** | [https://leetcode.com/problems/shortest-distance-to-a-character](https://www.google.com/url?q=https://leetcode.com/problems/shortest-distance-to-a-character&sa=D&source=editors&ust=1707482066925362&usg=AOvVaw0ch9OE4-vj9E78W8_QIoNm) | **Easy** |
| 408 | **Array, Greedy** | [https://leetcode.com/problems/can-place-flowers](https://www.google.com/url?q=https://leetcode.com/problems/can-place-flowers&sa=D&source=editors&ust=1707482066925495&usg=AOvVaw1wZ5kNnOYx_U8e0NKBluK_) | **Easy** |
| 409 | **Array, Hash Table, Sliding Window** | [https://leetcode.com/problems/contains-duplicate-ii](https://www.google.com/url?q=https://leetcode.com/problems/contains-duplicate-ii&sa=D&source=editors&ust=1707482066925634&usg=AOvVaw3Qmxl8x6Op1bAOsTEoJ_d7) | **Easy** |
| 410 | **Array, Hash Table, Math, Counting** | [https://leetcode.com/problems/number-of-good-pairs](https://www.google.com/url?q=https://leetcode.com/problems/number-of-good-pairs&sa=D&source=editors&ust=1707482066925746&usg=AOvVaw1fZcN0268Y7HzGEuKPPxr_) | **Easy** |
| 411 | **String, Stack, Greedy** | [https://leetcode.com/problems/minimum-add-to-make-parentheses-valid](https://www.google.com/url?q=https://leetcode.com/problems/minimum-add-to-make-parentheses-valid&sa=D&source=editors&ust=1707482066925843&usg=AOvVaw0JDUzJ8-q7RgvEBzUH0eVr) | **Medium** |
| 412 | **Array, Dynamic Programming, Greedy, Sorting** | [https://leetcode.com/problems/maximum-length-of-pair-chain](https://www.google.com/url?q=https://leetcode.com/problems/maximum-length-of-pair-chain&sa=D&source=editors&ust=1707482066925947&usg=AOvVaw023lWPUxxMwOsgsJu849-5) | **Medium** |
| 413 | **Depth-First Search, Breadth-First Search, Union Find, Graph** | [https://leetcode.com/problems/possible-bipartition](https://www.google.com/url?q=https://leetcode.com/problems/possible-bipartition&sa=D&source=editors&ust=1707482066926049&usg=AOvVaw1BC87jQ7R1xixEWQpIaaGr) | **Medium** |
| 414 | **Array, Hash Table, Greedy, Heap (Priority Queue)** | [https://leetcode.com/problems/split-array-into-consecutive-subsequences](https://www.google.com/url?q=https://leetcode.com/problems/split-array-into-consecutive-subsequences&sa=D&source=editors&ust=1707482066926159&usg=AOvVaw1YXJYhk47erlcYrU2vD8mz) | **Medium** |
| 415 | **Array, Depth-First Search, Breadth-First Search, Union Find, Matrix** | [https://leetcode.com/problems/making-a-large-island](https://www.google.com/url?q=https://leetcode.com/problems/making-a-large-island&sa=D&source=editors&ust=1707482066926267&usg=AOvVaw1eKHFedeS7EOw5EZpqUiBN) | **Hard** |
| 416 | **Array, Breadth-First Search, Matrix** | [https://leetcode.com/problems/shortest-path-in-binary-matrix](https://www.google.com/url?q=https://leetcode.com/problems/shortest-path-in-binary-matrix&sa=D&source=editors&ust=1707482066926437&usg=AOvVaw2emExFbWF9xygvX26hJIvD) | **Medium** |
| 417 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/count-good-nodes-in-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/count-good-nodes-in-binary-tree&sa=D&source=editors&ust=1707482066926605&usg=AOvVaw0eC0FXKyEdMdGQOUB51n_I) | **Medium** |
| 418 | **Array** | [https://leetcode.com/problems/shuffle-the-array](https://www.google.com/url?q=https://leetcode.com/problems/shuffle-the-array&sa=D&source=editors&ust=1707482066926774&usg=AOvVaw2W6dakO6nQGwbVUqW34kft) | **Easy** |
| 419 | **Array, Hash Table, Sliding Window, Heap (Priority Queue)** | [https://leetcode.com/problems/sliding-window-median](https://www.google.com/url?q=https://leetcode.com/problems/sliding-window-median&sa=D&source=editors&ust=1707482066927060&usg=AOvVaw3BB0vKxWw193yV9htmK3bR) | **Hard** |
| 420 | **Array, Binary Search** | [https://leetcode.com/problems/kth-missing-positive-number](https://www.google.com/url?q=https://leetcode.com/problems/kth-missing-positive-number&sa=D&source=editors&ust=1707482066927235&usg=AOvVaw0zmmaLoDreYAmisG4m0s49) | **Easy** |
| 421 | **Array, Binary Search** | [https://leetcode.com/problems/peak-index-in-a-mountain-array](https://www.google.com/url?q=https://leetcode.com/problems/peak-index-in-a-mountain-array&sa=D&source=editors&ust=1707482066927402&usg=AOvVaw1kmlHYNsZigsVXk7ezl2RH) | **Easy** |
| 422 | **Array, Linked List, Stack, Monotonic Stack** | [https://leetcode.com/problems/next-greater-node-in-linked-list](https://www.google.com/url?q=https://leetcode.com/problems/next-greater-node-in-linked-list&sa=D&source=editors&ust=1707482066927574&usg=AOvVaw3PJvGJae6Ye79gMvOiXqF5) | **Medium** |
| 423 | **Array, Dynamic Programming** | [https://leetcode.com/problems/minimum-swaps-to-make-sequences-increasing](https://www.google.com/url?q=https://leetcode.com/problems/minimum-swaps-to-make-sequences-increasing&sa=D&source=editors&ust=1707482066927745&usg=AOvVaw1Moi43-HIRuIzkzv-sISVZ) | **Hard** |
| 424 | **Array, Divide and Conquer, Sorting, Quickselect** | [https://leetcode.com/problems/wiggle-sort-ii](https://www.google.com/url?q=https://leetcode.com/problems/wiggle-sort-ii&sa=D&source=editors&ust=1707482066927935&usg=AOvVaw02zkmlLbKcg_obJIqrLd0Z) | **Medium** |
| 425 | **Array, Hash Table, Greedy, Sliding Window, Sorting, Heap (Priority Queue)** | [https://leetcode.com/problems/smallest-range-covering-elements-from-k-lists](https://www.google.com/url?q=https://leetcode.com/problems/smallest-range-covering-elements-from-k-lists&sa=D&source=editors&ust=1707482066928113&usg=AOvVaw3ZxiDxBli-fA7uSUlA0ZHl) | **Hard** |
| 426 | **Math, String, Recursion** | [https://leetcode.com/problems/integer-to-english-words](https://www.google.com/url?q=https://leetcode.com/problems/integer-to-english-words&sa=D&source=editors&ust=1707482066928283&usg=AOvVaw3L7TdEqWMlJUUG4hPCZ7Xe) | **Hard** |
| 427 | **Array, Dynamic Programming** | [https://leetcode.com/problems/partition-array-for-maximum-sum](https://www.google.com/url?q=https://leetcode.com/problems/partition-array-for-maximum-sum&sa=D&source=editors&ust=1707482066928462&usg=AOvVaw3B5i4zljDtpjG1jcFpsQ9w) | **Medium** |
| 428 | **Array, Sorting, Bucket Sort, Radix Sort** | [https://leetcode.com/problems/maximum-gap](https://www.google.com/url?q=https://leetcode.com/problems/maximum-gap&sa=D&source=editors&ust=1707482066928641&usg=AOvVaw3OSUI6W_uB76oZSbx2qhRE) | **Hard** |
| 429 | **Tree, Depth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/find-mode-in-binary-search-tree](https://www.google.com/url?q=https://leetcode.com/problems/find-mode-in-binary-search-tree&sa=D&source=editors&ust=1707482066928812&usg=AOvVaw30kTcDhkVZD2UeE1WPdtQl) | **Easy** |
| 430 | **Array, Hash Table, Binary Search, Dynamic Programming** | [https://leetcode.com/problems/longest-arithmetic-subsequence](https://www.google.com/url?q=https://leetcode.com/problems/longest-arithmetic-subsequence&sa=D&source=editors&ust=1707482066928987&usg=AOvVaw3kNNHDeQw_NpxkxeY9qJuJ) | **Medium** |
| 431 | **Array, Dynamic Programming** | [https://leetcode.com/problems/last-stone-weight-ii](https://www.google.com/url?q=https://leetcode.com/problems/last-stone-weight-ii&sa=D&source=editors&ust=1707482066929156&usg=AOvVaw0PxM-SdnF0dQ4qhaffN9CX) | **Medium** |
| 432 | **Array, Heap (Priority Queue)** | [https://leetcode.com/problems/last-stone-weight](https://www.google.com/url?q=https://leetcode.com/problems/last-stone-weight&sa=D&source=editors&ust=1707482066929334&usg=AOvVaw2huRO6m7-aNYPXHKO-1q6y) | **Easy** |
| 433 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/find-bottom-left-tree-value](https://www.google.com/url?q=https://leetcode.com/problems/find-bottom-left-tree-value&sa=D&source=editors&ust=1707482066929502&usg=AOvVaw3Z61-2tQgpYkLlurUkrwsE) | **Medium** |
| 434 | **Depth-First Search, Breadth-First Search, Union Find, Graph** | [https://leetcode.com/problems/regions-cut-by-slashes](https://www.google.com/url?q=https://leetcode.com/problems/regions-cut-by-slashes&sa=D&source=editors&ust=1707482066929684&usg=AOvVaw38kwF818zWo_twFId_qXIq) | **Medium** |
| 435 | **Array, Dynamic Programming, Backtracking, Bit Manipulation, Bitmask** | [https://leetcode.com/problems/beautiful-arrangement](https://www.google.com/url?q=https://leetcode.com/problems/beautiful-arrangement&sa=D&source=editors&ust=1707482066929858&usg=AOvVaw3JmbRHN61gzqw64ZIIcM9j) | **Medium** |
| 436 | **Array, Dynamic Programming, Matrix** | [https://leetcode.com/problems/minimum-falling-path-sum](https://www.google.com/url?q=https://leetcode.com/problems/minimum-falling-path-sum&sa=D&source=editors&ust=1707482066930029&usg=AOvVaw0iF6v_d-w7uBZ8fd6i9FF5) | **Medium** |
| 437 | **Array, Matrix, Simulation** | [https://leetcode.com/problems/diagonal-traverse](https://www.google.com/url?q=https://leetcode.com/problems/diagonal-traverse&sa=D&source=editors&ust=1707482066930198&usg=AOvVaw28l_BsB62kMsGGiidX3VdD) | **Medium** |
| 438 | **Array, Queue, Sliding Window, Heap (Priority Queue), Ordered Set, Monotonic Queue** | [https://leetcode.com/problems/longest-continuous-subarray-with-absolute-diff-less-than-or-equal-to-limit](https://www.google.com/url?q=https://leetcode.com/problems/longest-continuous-subarray-with-absolute-diff-less-than-or-equal-to-limit&sa=D&source=editors&ust=1707482066930386&usg=AOvVaw1NHRlHUB-_VNYX8H3S-SKj) | **Medium** |
| 439 | **Bit Manipulation** | [https://leetcode.com/problems/number-complement](https://www.google.com/url?q=https://leetcode.com/problems/number-complement&sa=D&source=editors&ust=1707482066930575&usg=AOvVaw1yg6Esv8sfNJu06Bz62ywz) | **Easy** |
| 440 | **Math, Binary Search** | [https://leetcode.com/problems/valid-perfect-square](https://www.google.com/url?q=https://leetcode.com/problems/valid-perfect-square&sa=D&source=editors&ust=1707482066930753&usg=AOvVaw1padtf5Y2g8LQmx69MP_tD) | **Easy** |
| 441 | **Math, Binary Search** | [https://leetcode.com/problems/arranging-coins](https://www.google.com/url?q=https://leetcode.com/problems/arranging-coins&sa=D&source=editors&ust=1707482066930925&usg=AOvVaw06bZDDjDTlxKGZDiBAaAgQ) | **Easy** |
| 442 | **Math, Two Pointers, String** | [https://leetcode.com/problems/next-greater-element-iii](https://www.google.com/url?q=https://leetcode.com/problems/next-greater-element-iii&sa=D&source=editors&ust=1707482066931102&usg=AOvVaw04FwWDr-6q1Wy0GNZE0MOM) | **Medium** |
| 443 | **Array, Binary Search, Depth-First Search, Breadth-First Search, Union Find, Heap (Priority Queue), Matrix** | [https://leetcode.com/problems/path-with-minimum-effort](https://www.google.com/url?q=https://leetcode.com/problems/path-with-minimum-effort&sa=D&source=editors&ust=1707482066931275&usg=AOvVaw1sZ3H6Z80aSea_0Uo3S24d) | **Medium** |
| 444 | **Array, String, Trie** | [https://leetcode.com/problems/search-suggestions-system](https://www.google.com/url?q=https://leetcode.com/problems/search-suggestions-system&sa=D&source=editors&ust=1707482066931466&usg=AOvVaw3sK6bTSmbWub1jA5tY5BTp) | **Medium** |
| 445 | **Array, Two Pointers, Matrix, Simulation** | [https://leetcode.com/problems/flipping-an-image](https://www.google.com/url?q=https://leetcode.com/problems/flipping-an-image&sa=D&source=editors&ust=1707482066931642&usg=AOvVaw2SDqeFGRzIjW3J3naJ1ODx) | **Easy** |
| 446 | **Tree, Design, Binary Search Tree, Heap (Priority Queue), Binary Tree, Data Stream** | [https://leetcode.com/problems/kth-largest-element-in-a-stream](https://www.google.com/url?q=https://leetcode.com/problems/kth-largest-element-in-a-stream&sa=D&source=editors&ust=1707482066931829&usg=AOvVaw378PLPawLcogfJi9iBUq9j) | **Easy** |
| 447 | **Array, Binary Search, Dynamic Programming, Matrix, Ordered Set** | [https://leetcode.com/problems/max-sum-of-rectangle-no-larger-than-k](https://www.google.com/url?q=https://leetcode.com/problems/max-sum-of-rectangle-no-larger-than-k&sa=D&source=editors&ust=1707482066932005&usg=AOvVaw1Ul1bPxMP48-wY3uLt-l5R) | **Hard** |
| 448 | **Hash Table, String, Binary Search, Design** | [https://leetcode.com/problems/time-based-key-value-store](https://www.google.com/url?q=https://leetcode.com/problems/time-based-key-value-store&sa=D&source=editors&ust=1707482066932215&usg=AOvVaw3iI__Slzw4TkYQG_fYdWX_) | **Medium** |
| 449 | **Linked List, Sorting** | [https://leetcode.com/problems/insertion-sort-list](https://www.google.com/url?q=https://leetcode.com/problems/insertion-sort-list&sa=D&source=editors&ust=1707482066932397&usg=AOvVaw0k7AHvunTNc7d2Mk5D8N8J) | **Medium** |
| 450 | **Two Pointers, String** | [https://leetcode.com/problems/string-compression](https://www.google.com/url?q=https://leetcode.com/problems/string-compression&sa=D&source=editors&ust=1707482066932568&usg=AOvVaw3V4WCleoaVNCHBp1b44vpU) | **Medium** |
| 451 | **Tree, Depth-First Search, Breadth-First Search** | [https://leetcode.com/problems/maximum-depth-of-n-ary-tree](https://www.google.com/url?q=https://leetcode.com/problems/maximum-depth-of-n-ary-tree&sa=D&source=editors&ust=1707482066932736&usg=AOvVaw1fTBiCNflFRltHsksP6FEl) | **Easy** |
| 452 | **Math** | [https://leetcode.com/problems/factorial-trailing-zeroes](https://www.google.com/url?q=https://leetcode.com/problems/factorial-trailing-zeroes&sa=D&source=editors&ust=1707482066932917&usg=AOvVaw0Uvn6-CZ0S1RVo9J_ykS__) | **Medium** |
| 453 | **Array, Sorting, Heap (Priority Queue), Simulation, Prefix Sum** | [https://leetcode.com/problems/car-pooling](https://www.google.com/url?q=https://leetcode.com/problems/car-pooling&sa=D&source=editors&ust=1707482066933098&usg=AOvVaw0WD-cmZ4M7-TZQaWpQhZM9) | **Medium** |
| 454 | **Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/find-largest-value-in-each-tree-row](https://www.google.com/url?q=https://leetcode.com/problems/find-largest-value-in-each-tree-row&sa=D&source=editors&ust=1707482066933273&usg=AOvVaw2X19WRUiEYkc_RbV209b8K) | **Medium** |
| 455 | **String, Dynamic Programming** | [https://leetcode.com/problems/shortest-common-supersequence](https://www.google.com/url?q=https://leetcode.com/problems/shortest-common-supersequence&sa=D&source=editors&ust=1707482066933458&usg=AOvVaw1OCkjIGNmVeTEcyrgCvxNX) | **Hard** |
| 456 | **Dynamic Programming** | [https://leetcode.com/problems/number-of-dice-rolls-with-target-sum](https://www.google.com/url?q=https://leetcode.com/problems/number-of-dice-rolls-with-target-sum&sa=D&source=editors&ust=1707482066933642&usg=AOvVaw0kscuwtAlTnumPcHjMyEwf) | **Medium** |
| 457 | **Hash Table, Linked List, Design, Heap (Priority Queue)** | [https://leetcode.com/problems/design-twitter](https://www.google.com/url?q=https://leetcode.com/problems/design-twitter&sa=D&source=editors&ust=1707482066933819&usg=AOvVaw2g4vsmNK9hiT57loUXwu8H) | **Medium** |
| 458 | **Array, Depth-First Search** | [https://leetcode.com/problems/array-nesting](https://www.google.com/url?q=https://leetcode.com/problems/array-nesting&sa=D&source=editors&ust=1707482066934002&usg=AOvVaw1HgFb9joWpDLVPseVPE9lY) | **Medium** |
| 459 | **Array, Math, Dynamic Programming, Game Theory** | [https://leetcode.com/problems/stone-game](https://www.google.com/url?q=https://leetcode.com/problems/stone-game&sa=D&source=editors&ust=1707482066934167&usg=AOvVaw3pNUK1kT1gl6r3WmiNAkIT) | **Medium** |
| 460 | **Array, Hash Table** | [https://leetcode.com/problems/degree-of-an-array](https://www.google.com/url?q=https://leetcode.com/problems/degree-of-an-array&sa=D&source=editors&ust=1707482066934347&usg=AOvVaw0ys7L-1mNVRIhg4rmjPkNJ) | **Easy** |
| 461 | **Hash Table, String, Bit Manipulation, Sorting** | [https://leetcode.com/problems/find-the-difference](https://www.google.com/url?q=https://leetcode.com/problems/find-the-difference&sa=D&source=editors&ust=1707482066934521&usg=AOvVaw2RCD0Jj3YYXqT8TCxK97qW) | **Easy** |
| 462 | **Tree, Depth-First Search, Breadth-First Search, Binary Search Tree, Binary Tree** | [https://leetcode.com/problems/minimum-absolute-difference-in-bst](https://www.google.com/url?q=https://leetcode.com/problems/minimum-absolute-difference-in-bst&sa=D&source=editors&ust=1707482066934702&usg=AOvVaw2_U4ldp1mOA0BASdE-a8Eu) | **Easy** |
| 463 | **Array, Binary Search, Matrix** | [https://leetcode.com/problems/count-negative-numbers-in-a-sorted-matrix](https://www.google.com/url?q=https://leetcode.com/problems/count-negative-numbers-in-a-sorted-matrix&sa=D&source=editors&ust=1707482066934882&usg=AOvVaw2hMeK0rB3fJarVRYVeT8mp) | **Easy** |
| 464 | **Array, Greedy, Heap (Priority Queue)** | [https://leetcode.com/problems/course-schedule-iii](https://www.google.com/url?q=https://leetcode.com/problems/course-schedule-iii&sa=D&source=editors&ust=1707482066935253&usg=AOvVaw203wVxQffkuxR8NGVVUsua) | **Hard** |
| 465 | **Hash Table, Tree, Depth-First Search, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/smallest-subtree-with-all-the-deepest-nodes](https://www.google.com/url?q=https://leetcode.com/problems/smallest-subtree-with-all-the-deepest-nodes&sa=D&source=editors&ust=1707482066935447&usg=AOvVaw1ZnLf3H0w4FaIJNXa5UqSP) | **Medium** |
| 466 | **Array, Matrix, Simulation** | [https://leetcode.com/problems/reshape-the-matrix](https://www.google.com/url?q=https://leetcode.com/problems/reshape-the-matrix&sa=D&source=editors&ust=1707482066935625&usg=AOvVaw24B62iuBa0ctKsp6CCT-Vq) | **Easy** |
| 467 | **Array, Matrix** | [https://leetcode.com/problems/toeplitz-matrix](https://www.google.com/url?q=https://leetcode.com/problems/toeplitz-matrix&sa=D&source=editors&ust=1707482066935793&usg=AOvVaw2MFOzffi-vdJQ15wbBIbFB) | **Easy** |
| 468 | **Dynamic Programming** | [https://leetcode.com/problems/knight-probability-in-chessboard](https://www.google.com/url?q=https://leetcode.com/problems/knight-probability-in-chessboard&sa=D&source=editors&ust=1707482066935963&usg=AOvVaw0W2riZmrQtjreBkx_dc-qw) | **Medium** |
| 469 | **Array, Two Pointers, Binary Search, Sorting** | [https://leetcode.com/problems/find-k-th-smallest-pair-distance](https://www.google.com/url?q=https://leetcode.com/problems/find-k-th-smallest-pair-distance&sa=D&source=editors&ust=1707482066936132&usg=AOvVaw2qPMB_XGMnGVuW3QGb3-an) | **Hard** |
| 470 | **Design, Segment Tree, Ordered Set** | [https://leetcode.com/problems/my-calendar-i](https://www.google.com/url?q=https://leetcode.com/problems/my-calendar-i&sa=D&source=editors&ust=1707482066936307&usg=AOvVaw1sai9yyYRWVmovBW0idYCy) | **Medium** |
| 471 | **String, Dynamic Programming** | [https://leetcode.com/problems/minimum-ascii-delete-sum-for-two-strings](https://www.google.com/url?q=https://leetcode.com/problems/minimum-ascii-delete-sum-for-two-strings&sa=D&source=editors&ust=1707482066936497&usg=AOvVaw3NXonkgtkYbG2xKqnx6Ef_) | **Medium** |
| 472 | **Array, Breadth-First Search, Matrix** | [https://leetcode.com/problems/shortest-path-in-a-grid-with-obstacles-elimination](https://www.google.com/url?q=https://leetcode.com/problems/shortest-path-in-a-grid-with-obstacles-elimination&sa=D&source=editors&ust=1707482066936675&usg=AOvVaw2snQP_Mh5cpPS8tFExy8XE) | **Hard** |
| 473 | **Array, Design, Prefix Sum** | [https://leetcode.com/problems/range-sum-query-immutable](https://www.google.com/url?q=https://leetcode.com/problems/range-sum-query-immutable&sa=D&source=editors&ust=1707482066936847&usg=AOvVaw3CB4M5I7LQD8qEo4hmD0d7) | **Easy** |
| 474 | **Tree, Breadth-First Search, Binary Tree** | [https://leetcode.com/problems/check-completeness-of-a-binary-tree](https://www.google.com/url?q=https://leetcode.com/problems/check-completeness-of-a-binary-tree&sa=D&source=editors&ust=1707482066937041&usg=AOvVaw3vh681s7RqjgaEWWJLKceq) | **Medium** |
| 475 | **Array** | [https://leetcode.com/problems/maximize-distance-to-closest-person](https://www.google.com/url?q=https://leetcode.com/problems/maximize-distance-to-closest-person&sa=D&source=editors&ust=1707482066937216&usg=AOvVaw1nF0lwikryceyuHJWcfDHV) | **Medium** |
| 476 | **Stack, Design, Queue** | [https://leetcode.com/problems/implement-stack-using-queues](https://www.google.com/url?q=https://leetcode.com/problems/implement-stack-using-queues&sa=D&source=editors&ust=1707482066937415&usg=AOvVaw1SsWTumFYnCx9u9NNM-Qjx) | **Easy** |
| 477 | **Math, Dynamic Programming, Bit Manipulation, Memoization, Game Theory, Bitmask** | [https://leetcode.com/problems/can-i-win](https://www.google.com/url?q=https://leetcode.com/problems/can-i-win&sa=D&source=editors&ust=1707482066937589&usg=AOvVaw2S5l6pDUXcoPbW5TvZX0pV) | **Medium** |
| 478 | **Hash Table, String, Sorting** | [https://leetcode.com/problems/custom-sort-string](https://www.google.com/url?q=https://leetcode.com/problems/custom-sort-string&sa=D&source=editors&ust=1707482066937756&usg=AOvVaw11HXfKHn-wMndwcta8oajr) | **Medium** |
| 479 | **Array, Hash Table** | [https://leetcode.com/problems/brick-wall](https://www.google.com/url?q=https://leetcode.com/problems/brick-wall&sa=D&source=editors&ust=1707482066937926&usg=AOvVaw3Mr-qInFnNbc6cKInb4u1h) | **Medium** |
| 480 | **Array, Hash Table, Stack, Monotonic Stack** | [https://leetcode.com/problems/next-greater-element-i](https://www.google.com/url?q=https://leetcode.com/problems/next-greater-element-i&sa=D&source=editors&ust=1707482066938099&usg=AOvVaw3UdsxsCmeDPCTTMQ-X4Wh0) | **Easy** |
| 481 | **Tree, Depth-First Search, Binary Tree** | [https://leetcode.com/problems/sum-of-root-to-leaf-binary-numbers](https://www.google.com/url?q=https://leetcode.com/problems/sum-of-root-to-leaf-binary-numbers&sa=D&source=editors&ust=1707482066938279&usg=AOvVaw2ff8a98HGX2geN9rkapQ9x) | **Easy** |
| 482 | **Depth-First Search, Breadth-First Search, Graph, Topological Sort** | [https://leetcode.com/problems/find-eventual-safe-states](https://www.google.com/url?q=https://leetcode.com/problems/find-eventual-safe-states&sa=D&source=editors&ust=1707482066938457&usg=AOvVaw2gRb3KYsswOz631agsZDGT) | **Medium** |
| 483 | **Array, Queue, Sorting, Simulation** | [https://leetcode.com/problems/reveal-cards-in-increasing-order](https://www.google.com/url?q=https://leetcode.com/problems/reveal-cards-in-increasing-order&sa=D&source=editors&ust=1707482066938624&usg=AOvVaw3k2eZnLNp0Kiycx6mVFZPo) | **Medium** |
| 484 | **Array, String, Dynamic Programming, Depth-First Search, Trie** | [https://leetcode.com/problems/concatenated-words](https://www.google.com/url?q=https://leetcode.com/problems/concatenated-words&sa=D&source=editors&ust=1707482066938798&usg=AOvVaw2hU_ASvO39u2mQKKX18Gcv) | **Hard** |
| 485 | **Array, Two Pointers, Greedy, Sorting** | [https://leetcode.com/problems/boats-to-save-people](https://www.google.com/url?q=https://leetcode.com/problems/boats-to-save-people&sa=D&source=editors&ust=1707482066938974&usg=AOvVaw0Qr0VJRMbPrDIclRObykSu) | **Medium** |
| 486 | **Array, Hash Table, Divide and Conquer, Tree, Binary Tree** | [https://leetcode.com/problems/construct-binary-tree-from-preorder-and-postorder-traversal](https://www.google.com/url?q=https://leetcode.com/problems/construct-binary-tree-from-preorder-and-postorder-traversal&sa=D&source=editors&ust=1707482066939150&usg=AOvVaw2lC4xt4ELiBIEsIEOuTn4i) | **Medium** |
| 487 | **Array, Dynamic Programming, Binary Indexed Tree** | [https://leetcode.com/problems/count-number-of-teams](https://www.google.com/url?q=https://leetcode.com/problems/count-number-of-teams&sa=D&source=editors&ust=1707482066939334&usg=AOvVaw3BgV_4ckaAOyqwd1yWU3Tg) | **Medium** |
| 488 | **Hash Table, String, Bit Manipulation, Sliding Window, Rolling Hash, Hash Function** | [https://leetcode.com/problems/repeated-dna-sequences](https://www.google.com/url?q=https://leetcode.com/problems/repeated-dna-sequences&sa=D&source=editors&ust=1707482066939512&usg=AOvVaw0tM5_WlXkcdy0KlfIvqT3I) | **Medium** |
| 489 | **Hash Table, String, Sliding Window** | [https://leetcode.com/problems/substring-with-concatenation-of-all-words](https://www.google.com/url?q=https://leetcode.com/problems/substring-with-concatenation-of-all-words&sa=D&source=editors&ust=1707482066939689&usg=AOvVaw0ExMkrK2Gs1f5YVtQrce4T) | **Hard** |
| 490 | **Array, Hash Table, Breadth-First Search** | [https://leetcode.com/problems/bus-routes](https://www.google.com/url?q=https://leetcode.com/problems/bus-routes&sa=D&source=editors&ust=1707482066939873&usg=AOvVaw0WZTBHph0WmzUrKYAhJxzg) | **Hard** |
| 491 | **Array, Hash Table, String** | [https://leetcode.com/problems/find-and-replace-pattern](https://www.google.com/url?q=https://leetcode.com/problems/find-and-replace-pattern&sa=D&source=editors&ust=1707482066940042&usg=AOvVaw34MVOXsElXmZjxrNSXy-VN) | **Medium** |
| 492 | **Array, Binary Search, Depth-First Search, Breadth-First Search, Union Find, Heap (Priority Queue), Matrix** | [https://leetcode.com/problems/swim-in-rising-water](https://www.google.com/url?q=https://leetcode.com/problems/swim-in-rising-water&sa=D&source=editors&ust=1707482066940214&usg=AOvVaw282kgFUasJYysDvQReO00B) | **Hard** |
| 493 | **Depth-First Search, Breadth-First Search, Union Find, Graph** | [https://leetcode.com/problems/number-of-operations-to-make-network-connected](https://www.google.com/url?q=https://leetcode.com/problems/number-of-operations-to-make-network-connected&sa=D&source=editors&ust=1707482066940397&usg=AOvVaw2IyzYK7o6yVrkhqhMiCZTY) | **Medium** |
| 494 | **Array, Divide and Conquer, Sorting, Heap (Priority Queue), Merge Sort, Bucket Sort, Radix Sort, Counting Sort** | [https://leetcode.com/problems/sort-an-array](https://www.google.com/url?q=https://leetcode.com/problems/sort-an-array&sa=D&source=editors&ust=1707482066940579&usg=AOvVaw3z2rHiiDpF0EeZJZ-ZluJA) | **Medium** |
| 495 | **Array, Backtracking, Matrix** | [https://leetcode.com/problems/path-with-maximum-gold](https://www.google.com/url?q=https://leetcode.com/problems/path-with-maximum-gold&sa=D&source=editors&ust=1707482066940755&usg=AOvVaw1L6vlBQOtz0Pmx6rl4t5_R) | **Medium** |
| 496 | **Array, Hash Table, Bit Manipulation, Sorting** | [https://leetcode.com/problems/set-mismatch](https://www.google.com/url?q=https://leetcode.com/problems/set-mismatch&sa=D&source=editors&ust=1707482066940925&usg=AOvVaw1bPa2FSu2dOTLmgsoxRqb8) | **Easy** |
| 497 | **Linked List** | [https://leetcode.com/problems/split-linked-list-in-parts](https://www.google.com/url?q=https://leetcode.com/problems/split-linked-list-in-parts&sa=D&source=editors&ust=1707482066941033&usg=AOvVaw3qbySPOKVgIygW7A391Bfb) | **Medium** |
| 498 | **Array, Two Pointers, Dynamic Programming, Enumeration** | [https://leetcode.com/problems/longest-mountain-in-array](https://www.google.com/url?q=https://leetcode.com/problems/longest-mountain-in-array&sa=D&source=editors&ust=1707482066941137&usg=AOvVaw1Q-jPo-9VRrnRR55uMpO3m) | **Medium** |
| 499 | **String, String Matching** | [https://leetcode.com/problems/rotate-string](https://www.google.com/url?q=https://leetcode.com/problems/rotate-string&sa=D&source=editors&ust=1707482066941238&usg=AOvVaw0L8icWmjzvjuGk0MR6jwYz) | **Easy** |
| 500 | **Array, Hash Table, String** | [https://leetcode.com/problems/unique-email-addresses](https://www.google.com/url?q=https://leetcode.com/problems/unique-email-addresses&sa=D&source=editors&ust=1707482066941348&usg=AOvVaw3ni5Vej5gtp36bPzaJwrY-) | **Easy** |