

CheatSheet: Leetcode Common Templates & Common Code Problems

INTERVIEW

- PDF Link: [cheatsheet-behavior-A4.pdf](#), Category: interview
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-leetcode-A4>
- Related posts: CheatSheet: System Design For Code Interview, [#denny-cheatsheets](#)

File me Issues or star this repo.

- CheatSheet: 30 Common Code Problems & Follow-ups

1.1 Top 25 Code Templates

| Num | Category/Tag | Example |
|-----|---|---|
| 1 | #bfs | Leetcode: Binary Tree Level Order Traversal |
| 2 | #dfs | Leetcode: Island Perimeter, Surrounded Regions |
| 3 | #binarysearch | Leetcode: Search Insert Position |
| 4 | #interval , #mergetwolist | Leetcode: Interval List Intersections |
| 5 | #twopointer , #array | Leetcode: Reverse Words in a String II |
| 6 | #twopointer | Leetcode: Two Sum |
| 7 | #backtracking , #subset | Leetcode: Subsets II |
| 8 | #linkedlist , #presum | Leetcode: Remove Zero Sum Consecutive Nodes from Linked List |
| 9 | #unionfind | Leetcode: Accounts Merge |
| 10 | #trie | Leetcode: Longest Word in Dictionary |
| 11 | #stack | Leetcode: Valid Parentheses |
| 12 | #stack | Leetcode: Reverse Substrings Between Each Pair of Parentheses |
| 13 | #heap | Leetcode: Top K Frequent Elements |
| 14 | #baseconversion | Leetcode: Base 7, Leetcode: Convert to Base -2 |
| 15 | #interval | Leetcode: Meeting Rooms II, Leetcode: My Calendar I |
| 16 | #monotone | Leetcode: Daily Temperatures |
| 17 | #knapsack | Leetcode: Coin Change |
| 18 | #sortbyfunction | Leetcode: Relative Sort Array |
| 19 | #slidingwindow | Leetcode: Longest Substring Without Repeating Characters |
| 20 | #editdistance , #dynamicprogramming | Leetcode: Longest Common Subsequence |
| 21 | #twopointer , #mergetwolist | Leetcode: Merge Sorted Array |
| 22 | #topologicalsort | Leetcode: Course Schedule |
| 23 | #bfs , bidirectional bfs | Leetcode: Word Ladder |
| 24 | #divideconquer , #recursive | |
| 25 | | |

1.2 Top 25 Graph Problems

| Num | Problem | Category/Tag | Summary |
|-----|--|---------------------------|-------------|
| 1 | Graph Connectivity: Count islands in a 2D matrix | #dfs, #unionfind | Leetcode: N |
| 2 | Get the size of the largest island | #dfs | Leetcode: M |
| 3 | Find shortest distance for two nodes in an undirected graph | #bfs | |
| 4 | Cycle detection in an undirected graph | | |
| 5 | Cycle detection in a directed graph | #topologicalsort | Leetcode: R |
| 6 | Detect all cycles in a directed graph | #dfs, #bfs | Leetcode: F |
| 7 | Whether a graph is a tree | #unionfind, #bfs | Leetcode: G |
| 8 | Minimum Spanning Tree(MST) of a weighted graph - Kruskal's algorithm | #unionfind | Leetcode: C |
| 9 | Shortest path for two nodes in a weighted graph - Dijkstra's algorithm | | |
| 10 | Find shortest paths in a weighted graph - Floyd-Warshall algorithm | #dfs, #dynamicprogramming | |
| 11 | Update a specific region | #dfs | Leetcode: F |
| 12 | Update regions for a given rule | | Leetcode: S |
| 13 | Number of Distinct Islands | #island, #dfs, #hashmap | Leetcode: N |
| 14 | Mark levels | | Leetcode: O |
| 15 | Duplicate edges | | Leetcode: R |
| 16 | Find a certain node in a graph | #unionfind | Leetcode: F |
| 17 | Coloring graph | #colorgraph, #bfs, #dfs | Leetcode: M |
| 18 | Find a certain path from source to destination in a graph | | Leetcode: P |
| 19 | Find the minimum steps from point1 to point2 | | Leetcode: W |
| 20 | Find all minimum paths from point1 to point2 | | Leetcode: W |
| 21 | All Paths from Source Lead to Destination | | Leetcode: A |
| 22 | Node connectivity problem for a sparse 2D matrix | #dfs, #bfs | Leetcode: E |
| 23 | Bricks Falling When Hit | #unionfind | Leetcode: B |
| 24 | Bridges in a connected graph - Tarjan's algorithm | | Leetcode: C |
| 25 | | | |

<https://cdn.dennyzhang.com/images/brain/dennyleetcode.png>

1.3 Top 5 Binarysearch Problems

| Num | Problem | Category/Tag | Summary |
|-----|-------------------------------------|---------------|--|
| 1 | Search Insert Position | #binarysearch | Leetcode: Search Insert Position, Leetcode: Time Based Key-Value |
| 2 | Find the first true | #binarysearch | Leetcode: First Bad Version |
| 3 | Find the last true | #binarysearch | |
| 4 | Binary search on monotonic function | #binarysearch | |
| 5 | | #binarysearch | |

1.4 Top 10 Dynamic Programming Problems

| Num | Problem | Time Complexity | Category/Tag | S |
|-----|---|-------------------------|--|---|
| 1 | Maximum subarray problem - Kadane's algorithm | O(n) | #maxsubarraysum, #dynamicprogramming | I |
| 2 | LIS - Longest increasing subsequence | O(n) | #lis, #string, #dynamicprogramming | I |
| 3 | LCS - Longest Common Subsequence | O(n*m) | #lcs, #editdistance, #dynamicprogramming | I |
| 4 | LPS - Longest Palindromic Subsequence | O(n) | #palindrome, #dynamicprogramming | I |
| 5 | Longest Palindromic Substring | O(n ²)/O(n) | #palindrome, #dynamicprogramming | I |
| 6 | Edit distance of two strings | O(n ²) | #editdistance, #dynamicprogramming | I |
| 7 | Maximum profits with certain costs | O(n ²) | #maxprofitwithcost, #dynamicprogramming | I |
| 8 | Regular Expression Matching | O(n*m) | #editdistance, #dynamicprogramming | I |
| 9 | Count of distinct subsequence | O(n) | #countdistinctmoves, #hashmap | I |
| 10 | Count out of boundary paths in a 2D matrix | O(n*m*N) | #countdistinctmoves, #bfs | I |

1.5 Top 10 BinaryTree Problems

| Num | Problem | Category/Tag | Summary |
|-----|---|--------------|---|
| 1 | Binary Tree Level Order Traversal | #bfs | Leetcode: Binary Tree Right Side View |
| 2 | Height of binary tree | #dfs | Leetcode: Balanced Binary Tree |
| 3 | LCA - Lowest Common Ancestor of a binary Tree | #dfs | Leetcode: Lowest Common Ancestor of a Binary Tree |
| 4 | Check whether a binary tree is a full binary tree | #dfs, #bfs | |
| 5 | Construct binary tree | #recursive | Leetcode: Construct Binary Tree from Preorder and Inorder Traversal |
| 6 | Right view of a tree | | |
| 7 | Validate Binary Search Tree | #dfs | Leetcode: Validate Binary Search Tree |

1.6 Top 5 String Problems

| Num | Problem | Category/Tag | Summary |
|-----|------------------------------|------------------------------------|--------------------------|
| 1 | Edit distance of two strings | #editdistance, #dynamicprogramming | Leetcode: Edit Distance |
| 2 | Remove duplicate letters | #stack, #greedy | Remove Duplicate Letters |
| 3 | Word ladder | #string, #bfs, #backtracking | Leetcode: Word Ladder |
| 4 | | | |
| 5 | | | |

1.7 Top 10 Math Problems

| Num | Problem | Category/Tag | Summary |
|-----|---|--------------|--|
| 1 | Check prime - Sieve of Eratosthenes | #prime | Leetcode: Count Primes |
| 2 | Check leap year | #leapyear | Leetcode: Day of the Week |
| 3 | gcd | #gcd | |
| 4 | Rectangle | #rectangle | |
| 5 | Rotate Array by k steps | #rotatelist | Leetcode: Rotate Array |
| 6 | Mapping data range of getRand algorithm | #random | Leetcode: Implement Rand10() Using Rand7() |

1.8 Top 5 Greedy Problems

| Num | Problem | Category/Tag | Summary |
|-----|---|---------------------------|---|
| 1 | Next Permutation | #nextpermutation, #greedy | Leetcode: Next Permutation |
| 2 | Split Array into Consecutive Subsequences | #splitarray, #greedy | Leetcode: Split Array into Consecutive Subsequences |
| 3 | Remove duplicate letters | #stack, #greedy | Remove Duplicate Letters |
| 4 | | | |
| 5 | | | |

1.9 Top 50 General Problems

| Num | Problem | Category/Tag | Example |
|-----|--|-----------------------------------|----------------------------|
| 1 | Longest substring with at most K distinct characters | #slidingwindow, #atmostkdistinct | Leetcode: Longest Substri |
| 2 | Longest subarray with maximum K 0s | #slidingwindow | Leetcode: Max Consecutiv |
| 3 | Seperate a list into several groups | #groupelements, #twopointer | Leetcode: Summary Rang |
| 4 | Split string | #string | Leetcode: License Key For |
| 5 | TopK problem | #heap, #topk | Leetcode: Top K Frequent |
| 6 | Longest Palindromic Subsequence | #dynamicprogramming | Leetcode: Longest Palindr |
| 7 | Sort one array based on another array | #sortbyfunction | Leetcode: Relative Sort A |
| 8 | Range update with lazy propagation | #combinedcaculation, #rangesum | Leetcode: Corporate Fligh |
| 9 | Monotone stack for consecutive subarrays | #montone | Leetcode: Online Stock Sp |
| 10 | Get all possibilities of subsets | #subset, #backtracking | Leetcode: Subsets II, Leet |
| 11 | Choose k numbers from a list | #combination, #backtracking | Leetcode: Combination Su |
| 12 | Combination from multiple segments | #combination, #backtracking | Leetcode: Letter Combina |
| 13 | Remove nodes from linked list | #linkedlist, #presum | Leetcode: Remove Zero Su |
| 14 | Check whether a linked list has a loop | | |
| 15 | Two pointers | #twosum, #twopointer | Leetcode: Two Sum |
| 16 | Buy stock for maximum profit list | #array, #greedy, #buystock | Leetcode: Best Time to B |
| 17 | Prefix search from a list of strings | #trie | Leetcode: Longest Word in |
| 18 | Factor Combinations | #combination, #backtracking | Leetcode: Factor Combina |
| 19 | Permutation without duplicates | #permutation, #backtracking | Leetcode: Palindrome Per |
| 20 | Int to string or string to int | #bitmanipulation | |
| 21 | Convert a number into negative base representation | #bitmanipulation, #baseconversion | Leetcode: Convert to Base |
| 22 | Network connectivity | #unionfind | Leetcode: Friend Circles |
| 23 | Build relationship among different sets | #unionfind | Leetcode: Accounts Merge |
| 24 | Knapsack problem to maximize benefits | #knapsack | Leetcode: Coin Change |
| 25 | Find the next greater value | #monotone | Leetcode: Daily Temperat |
| 26 | Meeting conflict | #interval | Leetcode: Meeting Rooms |
| 27 | Minimum conference rooms | #interval, #meetingconflict | Leetcode: Meeting Rooms |
| 28 | Quick slow pointers | #twopointer | LintCode: Middle of Link |
| 29 | Longest Repeating Character with at most K changes | #slidingwindow | Leetcode: Longest Repeat |
| 30 | Prefix and Suffix Search | #trie | Leetcode: Prefix and Suffi |
| 31 | Remove duplicate letters | #greedy, #string, #stack | Leetcode: Remove Duplica |
| 32 | Beautiful array | #divideconquer | Leetcode: Beautiful Array |
| 33 | Whether 132 pattern exists in array | #stack | Leetcode: 132 Pattern |
| 34 | Detect conflicts of intervals | #interval | Leetcode: Non-overlapping |
| 35 | Segment tree: solves range query problems quickly | #segmenttree | Leetcode: Range Sum Que |
| 36 | Find best meeting points for a list of nodes | #meetingpoint | Leetcode: Best Meeting P |
| 37 | Find the size of longest wiggle subsequence | #subsequence, #wiggle | Leetcode: Wiggle Subsequ |
| 38 | Sequence reconstruction | #topologicalsort | Leetcode: Sequence Recon |
| 39 | Construct Binary Tree from String | #stack | Construct Binary Tree fro |
| 40 | Use more space to save time | #stack | Leetcode: Min Stack |
| 41 | Min max game problems | #minmax, #dynamicprogramming | Leetcode: Predict the Win |
| 42 | Shortest Subarray with Sum at Least K | #monotone | Leetcode: Shortest Subarr |
| 43 | | | Travelling salesman proble |
| 44 | | | Leetcode: Remove Duplica |
| 45 | | | |
| 46 | | | |
| 47 | | | |

1.10 Tips: Think From The Other Direction

| Num | Name | Summary |
|-----|--|--|
| 1 | In graph, instead of deleting edges, add edge in reverse | Leetcode: Bricks Falling When Hit |
| 2 | Instead of BFS from empty to islands, do the otherwise | Leetcode: As Far from Land as Possible |
| 3 | Avoid deleting element from hashmaps | |

1.11 Common Tips For Clean Code

| Num | Name | Summary |
|-----|---|--|
| 1 | Calculate sum of a range quickly | #presum, Leetcode: Maximum Subarray |
| 2 | Move in four directions for a matrix | Leetcode: Sliding Puzzle |
| 3 | Split string by multiple separator | Leetcode: Brace Expansion |
| 4 | Add a dummy tailing element to simplify code | Leetcode: Brace Expansion |
| 5 | Fast slow pointers | Leetcode: Middle of Linked List |
| 6 | Deep copy an array | Leetcode: Combination Sum |
| 7 | Use arrays instead of hashmaps, if possible | Leetcode: Number of Days in a Month |
| 8 | Control the order of dfs | Leetcode: Subsets II |
| 9 | Avoid inserting into the head of an array | Leetcode: Path In Zigzag Labelled Binary Tree |
| 10 | From right to left, instead of left to right | Leetcode: Merge Sorted Array |
| 11 | Think the other way around | Add Items vs Remove Items, Increase Counter |
| 12 | Avoid unnecessary if...else... | res[i] = (diff/2 <= k), Leetcode: Can Make Palindrome |
| 13 | To get the case of K, solve: at most K - at most (K-1) | Leetcode: Subarrays with K Different Integers |
| 14 | Instead of deleting entry from hashmap, decrease counter | Leetcode: Longest Substring with At Most K Distinct Characters |
| 15 | Find the max/min; If not found, return 0 | Leetcode: Minimum Area Rectangle |
| 16 | With helper function vs without helper function | Leetcode: Longest Repeating Character Replacement |
| 17 | Instead of adding a character, try to delete one | Leetcode: Longest String Chain |
| 18 | #roundtrip: from left to right, then right to left | Leetcode: Shortest Distance to a Character |
| 19 | Delayed calculation to simplify the code | Leetcode: Interval List Intersections |
| 20 | Instead of removing, add padding elements | Leetcode: Duplicate Zeros |
| 21 | Initialize array with n+1 length to simplify code | Leetcode: Range Addition |
| 22 | Look for off-by-one errors, sometimes use i+1<len(l) vs i<len(l) | Leetcode: Previous Permutation With One Swap |
| 23 | Hashmap can reduce calculation, but may complicate things too | Leetcode: Maximum Frequency Stack |
| 24 | Sliding window to get the longest size of subarray | Leetcode: Max Consecutive Ones III |
| 25 | In matrix dfs, change cell to impossible value to avoid state hashmap | Leetcode: Word Search II |
| 26 | Avoid unnecessary precheck | |
| 27 | One pass instead of two pass | |
| 28 | Swiping line algorithm | |
| 29 | Add a dummy head node for linked list | |
| 30 | Hide details which are irrelevant | |

1.12 Golang Tips

| Name | Summary |
|---|---|
| Golang return a tuple | func dfs(root *TreeNode, max *float64) (sum int, cnt int), Leetcode: Binary Tree Maximum Path Sum |
| Use strings.Builder, instead of string | Leetcode: Unique Email Addresses |
| Variable Conversion | float64(x_int/y_int) != float64(x_int)/float64(y_int), Leetcode: Maximum Average Subarray II |
| For a list of objects, pass by value or reference | f(1 []*TreeNode) vs f(1 []*TreeNode), Leetcode: Lowest Common Ancestor of a Binary Tree |

1.13 Whiteboard Tips

| Name | Summary |
|---|--|
| Focus on your key motivations or thinkings | Pivot quickly from interviewers' feedback |
| Brute force algorithm add values | Intuitive algorithms are usually the starting points of optimal ones |
| Work through specific test case clearly | Reduce bugs, and help to obtain interviewers' feedback early |
| Naming variables could be tricky | Settle down a set of variables per your preference |
| You don't have to crack all problems/optimal algorithms | |

1.14 More Data Structure

| Name | Summary |
|----------------|---------|
| Tree map | |
| Inverted Index | |

1.15 Resource For Code Problems

| Name | Summary |
|----------------------|--|
| Leetcode summary | Link: Top Google Questions, Link: Top 100 Liked Questions, Link: Top Interview Questions |
| Leetcode summary | GitHub: kdn251/interviews, Github: Algorithms-and-Coding-Interviews |
| LeetCoder on YouTube | lee 215, Aoxiang Cui, happygirlzt |
| Online test websites | spoj.com, Google - codejam, hackerrank.com, hackerrank - hard, codeforces.com, poj.org |
| Online test websites | acm.hdu.edu.cn, acm.zju.edu.cn, acm.timus.ru, uva.onlinejudge.org |
| visualgo | visualising data structures and algorithms through animation |
| Reference | geeksforgeeks.org, Youtube: Abdul Bari - Algorithm |

1.16 More Resources

License: Code is licensed under MIT License.

<https://www.cs.princeton.edu/~rs/AlgsDS07/>

<https://www.geeksforgeeks.org/top-10-algorithms-in-interview-questions/>