

Microsoft DSA Previous Year Questions

Easy

- Write the code for this: Two Sum
(Given an array and a target, find two numbers that add up to the target.)
- Write the code for this: Replace All '?'s to Avoid Consecutive Repeating Characters
(In a string containing letters and '?', replace each '?' so no two identical letters are adjacent.)
- Write the code for this: Implement Queue using Stacks
(Use two stacks to simulate a FIFO queue with enqueue and dequeue operations.)
- Write the code for this: Maximum Subarray
(Find the contiguous subarray with the largest sum in a given array.)
- Write the code for this: Merge Sorted Array
(Merge two sorted arrays into one sorted array in-place.)
- Write the code for this: Palindrome Permutation
(Check if any permutation of a string can form a palindrome.)
- Write the code for this: Two Sum IV – Input is a BST
(Determine if there exist two nodes in a BST whose values sum to a given target.)
- Write the code for this: Min Stack
(Design a stack that supports push, pop, top, and retrieving the minimum element in $O(1)$.)

- Write the code for this: Best Time to Buy and Sell Stock
(Given an array of prices, compute the maximum profit from one buy-sell transaction.)
- Write the code for this: Merge Two Sorted Lists
(Merge two sorted linked lists into a single sorted list.)
- Write the code for this: Rectangle Overlap
(Given two axis-aligned rectangles, check if they overlap.)
- Write the code for this: Palindrome Linked List
(Check whether a singly linked list reads the same forwards and backwards.)
- Write the code for this: Kth Largest Element in a Stream
(Maintain the k^{th} largest element as new numbers are added to a data stream.)
- Write the code for this: Intersection of Two Linked Lists
(Find the node where two singly linked lists intersect, or return null.)
- Write the code for this: Valid Parentheses
(Validate that a string of '()', '[]', '{} is properly nested.)
- Write the code for this: Sqrt(x)
(Compute the integer square root of a non-negative integer.)
- Write the code for this: Find N Unique Integers Sum up to Zero
(Return any n distinct integers that add up to zero.)
- Write the code for this: Strobogrammatic Number
(Check if a number reads the same upside-down on a calculator display.)

- Write the code for this: Flood Fill
(Perform a “fill” on a 2D image starting from a pixel, replacing a color region.)
- Write the code for this: Fizz Buzz
(Print 1...n, replacing multiples of 3 with “Fizz”, 5 with “Buzz”, both with “FizzBuzz.”)
- Write the code for this: Ransom Note
(Given two strings, check if the first can be constructed by cutting letters from the second.)
- Write the code for this: Reverse String
(Reverse a character array in-place.)
- Write the code for this: First Unique Number
(Implement a data structure that returns the first unique number in a stream.)
- Write the code for this: Valid Anagram
(Determine if two strings are anagrams of each other.)
- Write the code for this: Plus One
(Given a large integer in array form, add one to it.)
- Write the code for this: Count Primes
(Count the number of prime numbers less than a given integer.)
- Write the code for this: Number of 1 Bits
(Count the number of ‘1’ bits in the binary representation of an unsigned integer.)
- Write the code for this: Valid Palindrome II

(Check if a string can become a palindrome by removing at most one character.)

- Write the code for this: Peak Index in a Mountain Array
(Find the peak (highest) index in an array where elements first strictly increase then decrease.)
- Write the code for this: Pascal's Triangle II
(Return the k^{th} row of Pascal's triangle.)
- Write the code for this: Nim Game
(Determine if the first player can win the Nim game with n stones.)
- Write the code for this: Keyboard Row
(Given words, return those that can be typed using letters of only one keyboard row.)
- Write the code for this: Goat Latin
(Convert a sentence to "Goat Latin" with a specific letter-shifting rule.)
- Write the code for this: DI String Match
(Given a pattern of 'I' and 'D', construct a permutation that fits it.)
- Write the code for this: Univalued Binary Tree
(Check if all nodes in a binary tree have the same value.)
- Write the code for this: Contains Duplicate
(Check if any value appears at least twice in an array.)
- Write the code for this: Move Zeroes
(Move all zeros in an array to the end while maintaining the order of non-zero elements.)
- Write the code for this: Intersection of Two Arrays

(Return the unique common elements of two integer arrays.)

Medium

- Write the code for this: LRU Cache
(Implement a cache with least-recently-used eviction in $O(1)$ time per operation.)
- Write the code for this: Maximum Length of a Concatenated String with Unique Characters
(Choose strings to concatenate so that all characters are distinct and length is maximized.)
- Write the code for this: Spiral Matrix
(Return all elements of an $m \times n$ matrix in spiral order.)
- Write the code for this: Boundary of Binary Tree
(List the boundary nodes of a binary tree in anti-clockwise order.)
- Write the code for this: Reverse Words in a String
(Reverse the order of words in a string, trimming extra spaces.)
- Write the code for this: Number of Islands
(Count connected "islands" of 1's in a 2D grid of 0's and 1's.)
- Write the code for this: Remove Comments
(Given lines of code, strip out C++-style `/**` and `/*...*/` comments.)
- Write the code for this: Design Tic-Tac-Toe
(Implement an $n \times n$ Tic-Tac-Toe game that can check for a win in $O(n)$.)

- Write the code for this: Populating Next Right Pointers in Each Node
(Connect each node in a perfect binary tree to its next right neighbor.)
- Write the code for this: Web Crawler
(Given a starting URL and fetch interface, crawl pages on the same hostname.)
- Write the code for this: Validate IP Address
(Determine if a string is a valid IPv4 or IPv6 address.)
- Write the code for this: Add Two Numbers II
(Add two numbers represented by linked lists without reversing them.)
- Write the code for this: Swap For Longest Repeated Character Substring
(Swap two characters in a string to maximize length of a repeated-character substring.)
- Write the code for this: Inorder Successor in BST II
(Given a node with parent pointers in a BST, return its in-order successor.)
- Write the code for this: Minimum Deletion Cost to Avoid Repeating Letters
(Delete characters from a string with given costs so no identical adjacent letters remain.)
- Write the code for this: Letter Combinations of a Phone Number
(Return all letter combinations that the digits of a phone number could represent.)
- Write the code for this: Spiral Matrix II
(Generate an $n \times n$ matrix filled in spiral order with $1 \dots n^2$.)

- Write the code for this: Validate Binary Search Tree
(Check whether a binary tree satisfies BST ordering properties.)
- Write the code for this: Reorder Routes to Make All Paths Lead to the City Zero
(Change road directions so all cities can reach city 0, minimizing changes.)
- Write the code for this: Longest Palindromic Substring
(Find the longest palindromic substring within a given string.)
- Write the code for this: Longest Substring with At Most K Distinct Characters
(Find the longest substring containing at most k distinct characters.)
- Write the code for this: Count Good Nodes in Binary Tree
(Count nodes whose value is $\geq$ all values on the path from the root.)
- Write the code for this: String Compression
(Compress a character array in-place by replacing repeated runs with character+count.)
- Write the code for this: Insert Delete GetRandom O(1)
(Support insert, delete, and getRandom on a set in constant time.)
- Write the code for this: Basic Calculator II
(Evaluate a string expression containing +, -, $\times$, $\div$ with proper precedence.)
- Write the code for this: Design Bounded Blocking Queue
(Implement a thread-safe queue with fixed capacity that blocks on full/empty.)
- Write the code for this: Android Unlock Patterns

(Count valid Android lock-screen patterns of length $m \dots n$ on a 3×3 grid.)

- Write the code for this: Add Two Numbers
(Add two numbers represented by linked lists in reverse order.)
- Write the code for this: Binary Tree Zigzag Level Order Traversal
(Return level order traversal of a tree, alternating direction each level.)
- Write the code for this: Lowest Common Ancestor of a Binary Tree
(Find the LCA of two nodes in a binary tree that isn't a BST.)
- Write the code for this: Permutations
(Generate all permutations of a list of distinct integers.)
- Write the code for this: Search a 2D Matrix
(Search for a target value in a matrix where each row and column is sorted.)
- Write the code for this: Reorganize String
(Rearrange letters so no two identical characters are adjacent, or report impossible.)
- Write the code for this: Search in Rotated Sorted Array
(Search a sorted array rotated at an unknown pivot for a target value.)
- Write the code for this: Group Anagrams
(Group words that are anagrams of each other.)
- Write the code for this: Design Hit Counter
(Record hits per timestamp and return number of hits in the past 5 minutes.)

- Write the code for this: Minimum Deletions to Make Character Frequencies Unique
(Delete the fewest chars so no two letters have the same frequency.)
- Write the code for this: Multiply Strings
(Multiply two non-negative integers given as strings, return the product as a string.)
- Write the code for this: Exclusive Time of Functions
(Compute each function's exclusive execution time given start/stop logs.)
- Write the code for this: Gas Station
(Determine if you can complete a circuit of gas stations given costs and supplies.)
- Write the code for this: Swap Nodes in Pairs
(Swap every two adjacent nodes in a linked list.)
- Write the code for this: Find the Duplicate Number
(Find the duplicate in an array containing $n+1$ integers of range $1 \dots n$.)
- Write the code for this: Construct Binary Tree from Preorder and Inorder Traversal
(Rebuild a binary tree given its preorder and inorder traversal lists.)
- Write the code for this: Largest Number
(Arrange a list of non-negative integers to form the largest possible number.)
- Write the code for this: Word Break
(Given a string and a dictionary, determine if it can be segmented into dictionary words.)



- Write the code for this: Course Schedule II
(Return a valid order to finish courses given prerequisites, or [] if impossible.)
- Write the code for this: Word Search
(Check if a given word exists in a 2D board of letters by moving horizontally or vertically.)
- Write the code for this: UTF-8 Validation
(Validate whether a byte sequence represents valid UTF-8 encoding.)
- Write the code for this: Number of Ways to Split a String
(Count ways to split a string into three parts with equal numbers of '1's.)
- Write the code for this: Minimize Rounding Error to Meet Target
(Round decimals up/down to integers so that their sum equals a target with minimal total error.)
- Write the code for this: Rotting Oranges
(Given fresh/rotten oranges in a grid, compute minutes until all fresh rot or return -1.)
- Write the code for this: Clone Graph
(Return a deep copy of an undirected graph given a reference node.)
- Write the code for this: String to Integer (atoi)
(Convert a string to a 32-bit signed integer following C's atoi rules.)
- Write the code for this: Battleships in a Board
(Count battleships on a grid—ships placed horizontally or vertically with no adjacency.)
- Write the code for this: Number of Dice Rolls With Target Sum

(Count ways to roll n dice to get a target sum.)

- Write the code for this: Longest Palindromic Subsequence
(Return the length of the longest palindromic subsequence in a string.)
- Write the code for this: Inorder Successor in BST
(Given a node in a BST, find its in-order successor.)
- Write the code for this: Max Area of Island
(Find the largest connected area of 1's in a grid.)
- Write the code for this: Longest Substring Without Repeating Characters
(Return the length of the longest substring with all unique characters.)
- Write the code for this: Encode and Decode TinyURL
(Implement a URL shortener: encode a long URL, then decode it back.)
- Write the code for this: Maximal Network Rank
(In a network of cities and roads, find the maximum rank of any pair of cities.)
- Write the code for this: Convert Binary Search Tree to Sorted Doubly Linked List
(Transform a BST into a sorted doubly linked list in-place.)
- Write the code for this: Coin Change II
(Given coin denominations and a total, count combinations to make that total.)
- Write the code for this: Populating Next Right Pointers in Each Node II

(Connect each node to its next right in any binary tree, not just perfect ones.)

- Write the code for this: Design Circular Queue
(Implement a circular queue with enqueue, dequeue, front, and rear operations.)
- Write the code for this: Task Scheduler
(Given tasks and a cooling interval, schedule them to minimize total time.)
- Write the code for this: Merge Intervals
(Merge overlapping intervals and return the merged list.)
- Write the code for this: Meeting Rooms II
(Given meeting time intervals, find the minimum number of conference rooms required.)
- Write the code for this: Pow(x, n)
(Implement x^n efficiently.)
- Write the code for this: Search a 2D Matrix II
(Search for a target in a matrix sorted row- and column-wise.)
- Write the code for this: Rotate Image
(Rotate an $n \times n$ matrix by 90° clockwise in-place.)
- Write the code for this: Rotate Array
(Rotate an array to the right by k steps.)
- Write the code for this: Binary Tree Level Order Traversal
(Return level order traversal of a binary tree.)
- Write the code for this: Next Closest Time

(Given a time “HH:MM”, find the next time using only the original digits.)

- Write the code for this: Reverse Linked List II
(Reverse a sublist of a linked list from position m to n.)
- Write the code for this: Max Stack
(Design a stack that supports push, pop, top, and retrieving the maximum element.)
- Write the code for this: Find the Celebrity
(Given a know(a,b) API, find the person who everyone knows but who knows no one.)
- Write the code for this: Binary Search Tree Iterator
(Implement an iterator over a BST that returns nodes in ascending order.)
- Write the code for this: First Missing Positive
(Find the smallest missing positive integer in an unsorted array.)
- Write the code for this: Trapping Rain Water
(Given heights, compute how much rainwater can be trapped.)
- Write the code for this: Word Search II
(Given a board and a list of words, find all words present on the board.)
- Write the code for this: Design Search Autocomplete System
(Given historical sentences with frequencies, suggest top-3 as you type.)
- Write the code for this: Boundary of Binary Tree
(List the tree’s boundary nodes in anti-clockwise order.)

- Write the code for this: Alien Dictionary
(Given a sorted dictionary in an alien language, derive the character order.)
- Write the code for this: Group Anagrams
(Group strings that are anagrams.)
- Write the code for this: Reorder Routes to Make All Paths Lead to City Zero
(Reorient edges so all nodes can reach node 0 with minimal changes.)
- Write the code for this: Parallel Courses II
(Given course prerequisites and a limit k per semester, find the minimum semesters.)
- Write the code for this: Basic Calculator II
(Evaluate a string with +, -, ×, ÷ and no parentheses.)
- Write the code for this: Minimum Deletions to Make Character Frequencies Unique
(Delete minimum chars so all character frequencies become unique.)
- Write the code for this: Rotate List
(Rotate a linked list to the right by k places.)
- Write the code for this: Partition List
(Partition a linked list around value x so all <x nodes come before ≥x.)
- Write the code for this: Flip Equivalent Binary Trees
(Check if two binary trees are the same up to flipping left/right children.)
- Write the code for this: Unique Paths

(Count paths from top-left to bottom-right in an $m \times n$ grid moving only down or right.)

Hard

- Write the code for this: **Palindrome Removal**
(Given an array, repeatedly remove palindromic subarrays to minimize operations.)
- Write the code for this: **Serialize and Deserialize N-ary Tree**
(Design methods to encode/decode an N-ary tree structure to/from a string.)
- Write the code for this: **Reverse Nodes in k-Group**
(Reverse every k nodes in a linked list.)
- Write the code for this: **Integer to English Words**
(Convert a non-negative integer to its English words representation.)
- Write the code for this: **Recover Binary Search Tree**
(Two nodes of a BST are swapped by mistake—fix the tree without changing its structure.)
- Write the code for this: **Merge k Sorted Lists**
(Merge k sorted linked lists into one sorted list efficiently.)
- Write the code for this: **Find Median from Data Stream**
(Maintain a data structure that returns the median of a stream of numbers.)
- Write the code for this: **LFU Cache**

(Design a cache that evicts the least-frequently used item in $O(1)$.)

- Write the code for this: Shortest Palindrome
(Add characters in front of a string to make it the shortest palindrome.)
- Write the code for this: Longest Consecutive Sequence
(Find the length of the longest run of consecutive integers in an array.)
- Write the code for this: Rearrange String k Distance Apart
(Reorder characters so the same letters are at least k apart, or report impossible.)
- Write the code for this: Trapping Rain Water
(Compute total water trapped after raining on elevation map.)
- Write the code for this: Basic Calculator
(Evaluate a string expression with +, -, parentheses, without using eval.)
- Write the code for this: Word Ladder
(Find the length of the shortest transformation sequence between two words.)
- Write the code for this: Minimum Window Substring
(Find the smallest substring of s containing all chars of t.)
- Write the code for this: Word Ladder II
(Return all shortest transformation sequences from beginWord to endWord.)
- Write the code for this: Count Unique Characters of All Substrings of a Given String

(Sum counts of unique characters across all substrings of a string.)

- Write the code for this: Maximal Rectangle
(Given a 2D binary matrix, find the largest rectangle of 1's.)
- Write the code for this: Sudoku Solver
(Fill a 9×9 board so each row, column, and 3×3 block contains 1–9.)
- Write the code for this: Alien Dictionary
(Derive the character order of an alien language given its sorted dictionary.)
- Write the code for this: Parallel Courses II
(Compute minimum semesters to finish all courses when you can take up to k at once.)
- Write the code for this: Design Skiplist
(Implement a skip list supporting add, erase, and search in $O(\log n)$.)
- Write the code for this: Text Justification
(Format text to fully justify with a given line width.)
- Write the code for this: Expression Add Operators
(Insert +, −, × between digits to evaluate to a target value.)
- Write the code for this: All Nodes Distance K in Binary Tree
(Return all nodes at distance K from a target node in a binary tree.)
- Write the code for this: N-Queens
(Place n queens on an $n \times n$ board so none attack each other.)
- Write the code for this: Count of Smaller Numbers After Self
(For each element, count how many smaller elements appear to its right.)

- Write the code for this: Russian Doll Envelopes
(Find the max number of envelopes you can nest by width and height.)
- Write the code for this: Binary Tree Maximum Path Sum
(Find a path yielding the largest sum of node values in a binary tree.)
- Write the code for this: Evaluate Division
(Given equations as $a/b = k$, answer queries of the form x/y .)
- Write the code for this: Design Search Autocomplete System
(Suggest top-3 historical sentences as you type each character.)
- Write the code for this: Design Skiplist
(Implement a skip list with add, remove, and search in expected $O(\log n)$.)
- Write the code for this: LFU Cache
(Design a cache that evicts the least-frequently used key in $O(1)$.)
- Write the code for this: Recover Binary Search Tree
(Fix a BST where two nodes are swapped by mistake.)
- Write the code for this: Minimum Window Subsequence
(Find the shortest subsequence of S that contains T as a subsequence.)
- Write the code for this: Sliding Window Maximum
(Return max for each sliding window of size k over an array.)
- Write the code for this: Sliding Puzzle
(Solve the 2×3 sliding puzzle, returning the minimum moves or -1 .)

- Write the code for this: Word Search II
(Find all words from a list that can be formed on a letter board.)
- Write the code for this: Median of Two Sorted Arrays
(Find the median of two sorted arrays in $O(\log(m+n))$.)
- Write the code for this: Minimum Knight Moves
(Compute the fewest moves a knight needs to reach a given square.)
- Write the code for this: Basic Calculator
(Evaluate a string with +, -, parentheses under more stringent constraints.)
- Write the code for this: Max Points on a Line
(Find the maximum number of points that lie on the same straight line.)
- Write the code for this: Palindrome Removal
(Remove palindromic subarrays to clear the array in minimum moves.)
- Write the code for this: Design Search Autocomplete System
(Given sentence history, suggest top-3 matches as you type.)
- Write the code for this: Trapping Rain Water
(Calculate water trapped in 1D elevation map.)
- Write the code for this: LFU Cache
(Evict least-frequently used cache item in $O(1)$.)