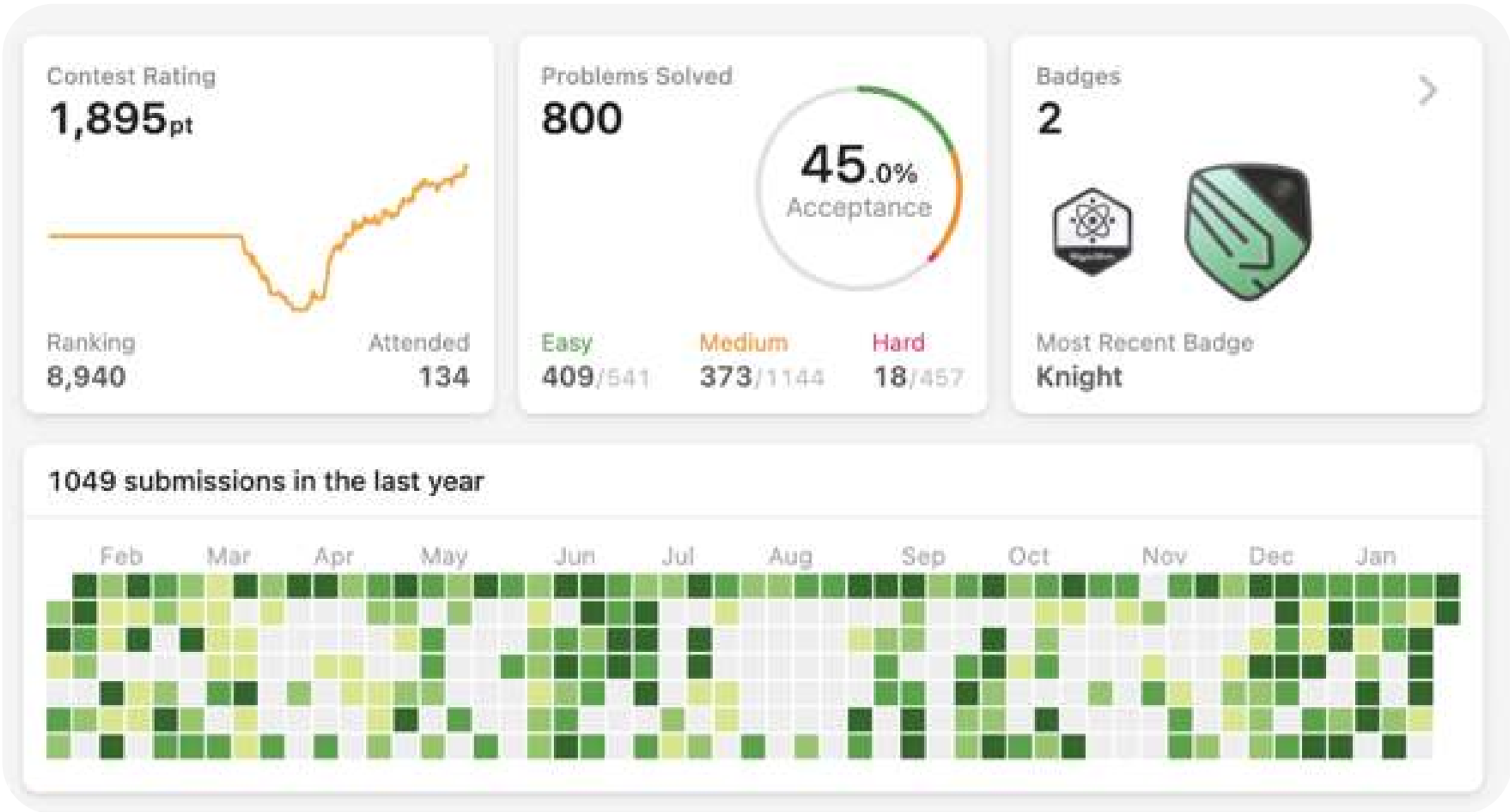
100

MUST DO

LEETCODE

PROBLEMS



Disclaimer

Everyone learns uniquely.

What matters is developing the problem solving ability to solve new problems.

This Doc will help you with the same.

1. ARRAYS:

1. Given an integer array nums, return an array answer such that answer[i] is equal to the product of all the elements of nums except nums[i].

   Practice

2. Given an integer array nums, move all 0's to the end of it while maintaining the relative order of the non-zero elements. Note that you must do this in-place without making a copy of the array.

   Practice

3. You are given an array of prices where prices[i] is the price of a given stock on an ith day. You want to maximise your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock. Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.

   Practice

4. The next permutation of an array of integers is the next lexicographically greater permutation of its integer. Given an array of integers nums, find the next permutation of nums.
   The replacement must be in place and use only constant extra memory.

   Practice

5. Given an array of integers and an integer target, return indices of the two numbers such that they add up to target.

   Practice

6. You are given an integer array height of length n. There are n vertical lines drawn such that the two endpoints of the ith line are (i, 0) and (i, height[i]). Find two lines that together with the x-axis form a container, such that the container contains the most water. Return the maximum amount of water a container can store.

   Practice

7. Given an array nums with n objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue.

   Practice

8. Given an array of positive integers nums and a positive integer target, return the minimal length of a subarray whose sum is greater than or equal to target. If there is no such subarray, return 0 instead.

   Practice

9. You are given two integer arrays nums1 and nums2, sorted in non-decreasing order, and two integers m and n, representing the number of elements in nums1 and nums2 respectively. Merge nums1 and nums2 into a single array sorted in non-decreasing order.

Practice

2. STRINGS:

1. Write a function that reverses a string. The input string is given as an array of characters s. You must do this by modifying the input array in-place with O(1) extra memory.

   Practice

2. Given two strings s and p, return an array of all the start indices of p's anagrams in s.

   Practice

3. Given two strings s and t of lengths m and n respectively, return the minimum window substring of s such that every character in t (including duplicates) is included in the window. If there is no such substring, return the empty string "".

   Practice

4. You are given a string s and an integer k. You can choose any character of the string and change it to any other uppercase English character. You can perform this operation at most k times.

   Practice

5. Given an array of strings strs, group the anagrams together. You can return the answer in any order. An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

   Practice

6. Given a Roman numeral, convert it to an integer

   Practice

7. Given a string s, sort it in decreasing order based on the frequency of the characters. The frequency of a character is the number of times it appears in the string. Return the sorted string. If there are multiple answers, return any of them.

   Practice

8. Given a string s of '(' , ')' and lowercase English characters. Your task is to remove the minimum number of parentheses ('(' or ')', in any positions) so that the resulting parentheses string is valid and return any valid string.

   Practice

9. Given two strings s1 and s2, return true if s2 contains a permutation of s1, or false otherwise.
   In other words, return true if one of s1's permutations is the substring of s2.

   Practice

3. SEARCHING AND SORTING ALGORITHMS:

1. Given an array of integers nums which is sorted in ascending order, and an integer target, write a function to search target in nums. If target exists, then return its index. Otherwise, return -1. You must write an algorithm with O(log n) runtime complexity.

   Practice

2. Given an n x n matrix where each of the rows and columns is sorted in ascending order, return the kth smallest element in the matrix.
   Note that it is the kth smallest element in the sorted order, not the kth distinct element.
   You must find a solution with a memory complexity better than O(n2).

   Practice

3. Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of the two sorted arrays.
   The overall run time complexity should be O(log (m+n)).

   Practice

4. Given an integer array nums and an integer k, return the kth largest element in the array.
   Note that it is the kth largest element in the sorted order, not the kth distinct element.
   You must solve it in O(n) time complexity.

   Practice

4. RECURSION:

1. Given n pairs of parentheses, write a function to generate all combinations of well-formed parentheses.

   Practice

2. Given an integer array nums of unique elements, return all possible Subsets (the power set).
   The solution set must not contain duplicate subsets. Return the solution in any order

   Practice

3. Given an array nums of distinct integers, return all the possible permutations. You can return the answer in any order.

   Practice

4. Given a string s, partition s such that every substring of the partition is a palindrome
   Return all possible palindrome partitioning of s.

   Practice

5. Given a collection of candidate numbers (candidates) and a target number (target), find all unique combinations in candidates where the candidate numbers sum to target.
   Each number in candidates may only be used once in the combination.
   Note: The solution set must not contain duplicate combinations.

   Practice

6. Given the head of a linked list and an integer val, remove all the nodes of the linked list that has Node.val == val, and return the new head.

   Practice

7. Given a string expression of numbers and operators, return all possible results from computing all the different possible ways to group numbers and operators. You may return the answer in any order.

   Practice

5. HASHING:

1. Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.
   You may assume that each input would have exactly one solution, and you may not use the same element twice.
   You can return the answer in any order.

   Practice

2. Given an unsorted integer array nums, return the smallest missing positive integer.
   You must implement an algorithm that runs in O(n) time and uses constant extra space.

   Practice

3. Design a data structure that follows the constraints of a Least Recently Used (LRU) cache.

   Practice

4. You are given a 0-indexed, strictly increasing integer array nums and a positive integer diff. A triplet (i, j, k) is an arithmetic triplet if the following conditions are met:
   i < j < k,
   nums[j] - nums[i] == diff, and
   nums[k] - nums[j] == diff.
   Return the number of unique arithmetic triplets.

Practice

6. MATRICES AND MULTIDIMENSIONAL ARRAYS:

1. Given an m x n matrix, return all elements of the matrix in spiral order.

   Practice

2. Given an m x n integer matrix matrix, if an element is 0, set its entire row and column to 0's.
   You must do it in place.

   Practice

3. Determine if a 9 × 9 Sudoku board is valid. Only the filled cells need to be validated according to the following rules:
   Each row must contain the digits 1-9 without repetition.
   Each column must contain the digits 1-9 without repetition.
   Each of the nine 3 × 3 sub-boxes of the grid must contain the digits 1-9 without repetition.

   Practice

4. You are given an n x n 2D matrix representing an image, rotate the image by 90 degrees (clockwise).
   You have to rotate the image in-place, which means you have to modify the input 2D matrix directly. DO NOT allocate another 2D matrix and do the rotation.

   Practice

5. Write an efficient algorithm that searches for a value target in an m x n integer matrix matrix. This matrix has the following properties:
   Integers in each row are sorted from left to right.
   The first integer of each row is greater than the last integer of the previous row.

   Practice

6. Given an m x n grid of characters board and a string word, return true if word exists in the grid.
   The word can be constructed from letters of sequentially adjacent cells, where adjacent cells are horizontally or vertically neighboring. The same letter cell may not be used more than once.

   Practice

7. Given an m x n binary matrix mat, return the distance of the nearest 0 for each cell. The distance between two adjacent cells is 1.

   Practice

7. LINKED LIST:

1. Given the head of a singly linked list, reverse the list, and return the reversed list.

   Practice

2. Given the head of a linked list, rotate the list to the right by k places.

   Practice

3. Given head, the head of a linked list, determine if the linked list has a cycle in it.
   There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the next pointer. Internally, pos is used to denote the index of the node that tail's next pointer is connected to. Note that pos is not passed as a parameter.
   Return true if there is a cycle in the linked list. Otherwise, return false.

   Practice

4. You are given the heads of two sorted linked lists list1 and list2.
   Merge the two lists in a one sorted list. The list should be made by splicing together the nodes of the first two lists.
   Return the head of the merged linked list.

   Practice

5. Given the head of a linked list, remove the nth node from the end of the list and return its head.

   Practice

6. You are given the head of a singly linked-list. The list can be represented as:

   L0 → L1 → … → Ln - 1 → Ln

   Reorder the list to be on the following form:

   L0 → Ln → L1 → Ln - 1 → L2 → Ln - 2 → …

   You may not modify the values in the list's nodes. Only nodes themselves may be changed.

   Practice

7. Given the heads of two singly linked-lists headA and headB, return the node at which the two lists intersect. If the two linked lists have no intersection at all, return null.

   Practice

8. Given the head of a linked list, reverse the nodes of the list k at a time, and return the modified list. k is a positive integer and is less than or equal to the length of the linked list. If the number of nodes is not a multiple of k then left-out nodes, in the end, should remain as it is.
   You may not alter the values in the list's nodes, only nodes themselves may be changed.

   Practice

9. Given the head of a singly linked list, return true if it is a palindrome or false otherwise.

   Practice

10. Given the root of a binary tree, flatten the tree into a "linked list":
    The "linked list" should use the same TreeNode class where the right child pointer points to the next node in the list and the left child pointer is always null.
    The "linked list" should be in the same order as a pre-order traversal of the binary tree.

    Practice

8. BIT MANIPULATION & MATH CONCEPTS:

1. Given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array.

   Practice

2. Given an integer n, return an array ans of length n + 1 such that for each i (0 <= i <= n), ans[i] is the number of 1's in the binary representation of i.

   Practice

3. Given a non-empty array of integers nums, every element appears twice except for one. Find that single one.
   You must implement a solution with a linear runtime complexity and use only constant extra space.

   Practice

4. Given two integer arrays nums1 and nums2, return an array of their intersection. Each element in the result must appear as many times as it shows in both arrays and you may return the result in any order.

   Practice

5. You are given two strings s and t. String t is generated by random shuffling string s and then add one more letter at a random position. Return the letter that was added to t.

Practice

9. STACKS AND QUEUES:

1. Implement a first in first out (FIFO) queue using only two stacks. The implemented queue should support all the functions of a normal queue (push, peek, pop, and empty).

   Practice

2. Implement a last-in-first-out (LIFO) stack using only two queues. The implemented stack should support all the functions of a normal stack (push, top, pop, and empty).

   Practice

3. Given an array of integers temperatures represents the daily temperatures, return an array answer such that answer[i] is the number of days you have to wait after the ith day to get a warmer temperature. If there is no future day for which this is possible, keep answer[i] == 0 instead.

   Practice

4. Given an array of integers heights representing the histogram's bar height where the width of each bar is 1, return the area of the largest rectangle in the histogram.

   Practice

5. Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining.

   Practice

6. Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

   Practice

7. Design your implementation of the circular queue. The circular queue is a linear data structure in which the operations are performed based on FIFO (First In First Out) principle, and the last position is connected back to the first position to make a circle. It is also called "Ring Buffer".

   Practice

8. You are given an array of integers nums, there is a sliding window of size k which is moving from the very left of the array to the very right. You can only see the k numbers in the window. Each time the sliding window moves right by one position.
   Return the max sliding window.

   Practice

9. Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.

   Practice

10. You are given an m x n grid where each cell can have one of three values:
    0 representing an empty cell,
    1 representing a fresh orange, or
    2 representing a rotten orange.
    Every minute, any fresh orange that is 4-directionally adjacent to a rotten orange becomes rotten.
    Return the minimum number of minutes that must elapse until no cell has a fresh orange. If this is impossible, return -1.

    Practice

10. TREES & BINARY SEARCH TREES:

1. Given the root of a binary tree, return its maximum depth. A binary tree's maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

   Practice

2. Given the root of a binary tree, invert the tree, and return its root.

   Practice

3. A path in a binary tree is a sequence of nodes where each pair of adjacent nodes in the sequence has an edge connecting them. A node can only appear in the sequence at most once. Note that the path does not need to pass through the root. The path sum of a path is the sum of the node's values in the path. Given the root of a binary tree, return the maximum path sum of any non-empty path.

   Practice

4. Given the roots of two binary trees p and q, write a function to check if they are the same or not. Two binary trees are considered the same if they are structurally identical, and the nodes have the same value.

   Practice

5. Given the root of a binary tree, return the level order traversal of its nodes' values. (i.e., from left to right, level by level).

   Practice

6. Given a binary tree, find the lowest common ancestor (LCA) of two given nodes in the tree.

   Practice

7. Given two integer arrays preorder and inorder where preorder is the preorder traversal of a binary tree and inorder is the inorder traversal of the same tree, construct and return the binary tree.

   Practice

8. Given the root of a binary tree, imagine yourself standing on the right side of it, return the values of the nodes you can see ordered from top to bottom.

   Practice

9. Given the root of a binary tree, determine if it is a valid binary search tree (BST).

   Practice

10. Given the root of a binary search tree, and an integer k, return the kth smallest value (1-indexed) of all the values of the nodes in the tree.

    Practice

11. Given the root of a binary tree, return the zigzag level order traversal of its nodes' values. (i.e., from left to right, then right to left for the next level and alternate between).

    Practice

12. Given a binary tree, determine if it is height-balanced

    Practice

11. TRIES:

1. A trie (pronounced as "try") or prefix tree is a tree data structure used to efficiently store and retrieve keys in a dataset of strings. There are various applications of this data structure, such as autocomplete and spellchecker. Implement the Trie.

   Practice

2. Design a data structure that supports adding new words and finding if a string matches any previously added string.

   Practice

3. Given an integer array nums, return the maximum result of nums[i] XOR nums[j], where 0 <= i <= j < n.

   Practice

4. Given a string s and a dictionary of strings wordDict, return true if s can be segmented into a space-separated sequence of one or more dictionary words. Note that the same word in the dictionary may be reused multiple times in the segmentation.

   Practice

5. Given an m x n board of characters and a list of strings words, return all words on the board. Each word must be constructed from letters of sequentially adjacent cells, where adjacent cells are horizontally or vertically neighboring. The same letter cell may not be used more than once in a word.

Practice

12. HEAPS:

1. You are given an array of k linked-lists lists, each linked-list is sorted in ascending order. Merge all the linked-lists into one sorted linked-list and return it.

   Practice

2. Given an integer array nums and an integer k, return the k most frequent elements. You may return the answer in any order.

   Practice

3. Given a string s, rearrange the characters of s so that any two adjacent characters are not the same. Return any possible rearrangement of s or return "" if not possible.

   Practice

4. The median is the middle value in an ordered integer list. If the size of the list is even, there is no middle value, and the median is the mean of the two middle values. Implement the MedianFinder class:

   Practice

13. GRAPHS:

1. Given an m x n 2D binary grid grid which represents a map of '1's (land) and '0's (water), return the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

   Practice

2. Given is a 2D adjacency list representation of a graph. Check whether the graph is a Bipartite graph.

   Practice

3. An image is represented by an m x n integer grid image where image[i][j] represents the pixel value of the image. You are also given three integers sr, sc, and color. You should perform a flood fill on the image starting from the pixel image[sr][sc].

   Practice

4. Given an m x n integers matrix, return the length of the longest increasing path in matrix. From each cell, you can either move in four directions: left, right, up, or down. You may not move diagonally or move outside the boundary (i.e., wrap-around is not allowed).

   Practice

5. There are n cities connected by some number of flights. You are given an array flights where flights[i] = [fromi, toi, pricei] indicates that there is a flight from city fromi to city toi with cost pricei. You are also given three integers src, dst, and k, return the cheapest price from src to dst with at most k stops. If there is no such route, return -1.

   Practice

6. There are n cities. Some of them are connected, while some are not. If city a is connected directly with city b, and city b is connected directly with city c, then city a is connected indirectly with city c.

   A province is a group of directly or indirectly connected cities and no other cities outside of the group.
   You are given an n x n matrix isConnected where isConnected[i][j] = 1 if the ith city and the jth city are directly connected, and isConnected[i][j] = 0 otherwise. Return the total number of provinces.

   Practice

14. DYNAMIC PROGRAMMING AND GREEDY:

1. Given an integer array nums, find a subarray that has the largest product, and return the product.

   Practice

2. Given an integer array nums, return the length of the longest strictly increasing subsequence

   Practice

3. Given two strings word1 and word2, return the minimum number of operations required to convert word1 to word2.
   You have the following three operations permitted on a word:
   Insert a character
   Delete a character
   Replace a character

   Practice

4. You are given an integer array coins representing coins of different denominations and an integer amount representing a total amount of money. Return the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combination of the coins, return -1. You may assume that you have an infinite number of each kind of coin.

   Practice

5. Given a non-empty array nums containing only positive integers, find if the array can be partitioned into two subsets such that the sum of elements in both subsets is equal.

   Practice

6. Given two strings text1 and text2, return the length of their longest common subsequence. If there is no common subsequence, return 0.

   Practice

8. Given a string s and a dictionary of strings wordDict, return true if s can be segmented into a space-separated sequence of one or more dictionary words.

   Practice

7. There is a robot on an m x n grid. The robot is initially located at the top-left corner (i.e., grid[0][0]). The robot tries to move to the bottom-right corner (i.e., grid[m - 1][n - 1]). The robot can only move either down or right at any point in time. Given the two integers m and n, return the number of possible unique paths that the robot can take to reach the bottom-right corner.

Practice