Project2——被测项目问题描述

说明：每一个问题对应的源代码文件名与问题名称一致。

1. Add Two Numbers

You are given two linked lists representing two non-negative numbers. The digits are stored in reverse order and each of their nodes contains a single digit. Add the two numbers and return it as a linked list.

2. Add Two Numbers 2

You are given two linked lists representing two non-negative numbers. The most significant digit comes first and each of their nodes contains a single digit. Add the two numbers and return it as a linked list. You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example:
Input: (7 -> 2 -> 4 -> 3) + (5 -> 6 -> 4)
Output: 7 -> 8 -> 0 -> 7

3. AlienAlphabet

Given dictionary of alien language. You need to find order of alphabets based on dictionary

4. Asteroid Collision

We are given an array of integers representing asteroids in a row. For each asteroid, the absolute value represents its size, and the sign represents its direction (positive meaning right, negative meaning left). Each asteroid moves at the same speed.
Find out the state of the asteroids after all collisions. If two asteroids meet, the smaller one will explode. If both are the same size, both will explode. Two asteroids moving in the same direction will never meet.

Example 1:
Input: asteroids = [5, 10, -5]
Output: [5, 10]
Explanation: The 10 and -5 collide resulting in 10. The 5 and 10 never collide.

Example 2:
Input: asteroids = [8, -8]
Output: []
Explanation: The 8 and -8 collide exploding each other.

Example 3:
Input:
asteroids = [10, 2, -5]
Output: [10]
Explanation: The 2 and -5 collide resulting in -5. The 10 and -5 collide resulting in

10.

5. BattleShipsInBoard

Given an 2D board, count how many different battleships are in it. The battleships are represented with 'X's, empty slots are represented with '.'s. You may assume the following rules:

You receive a valid board, made of only battleships or empty slots.

Battleships can only be placed horizontally or vertically. In other words, they can only be made of the shape 1xN (1 row, N columns) or Nx1 (N rows, 1 column), where N can be of any size.

At least one horizontal or vertical cell separates between two battleships - there are no adjacent battleships.

Example:

```
X..X
...X
...X
```

In the above board there are 2 battleships.

Invalid Example:

```
...X
XXXX
...X
```

This is an invalid board that you will not receive - as battleships will always have a cell separating between them.

6. Beautiful Arrangement

Suppose you have N integers from 1 to N. We define a beautiful arrangement as an array that is constructed by these N numbers successfully if one of the following is true for the ith position (1 <= i <= N) in this array:

The number at the ith position is divisible by i.i is divisible by the number at the ith position.

Now given N, how many beautiful arrangements can you construct?

Example 1:

Input: 2   Output: 2

Explanation:

The first beautiful arrangement is [1, 2]:

Number at the 1st position (i=1) is 1, and 1 is divisible by i (i=1).

Number at the 2nd position (i=2) is 2, and 2 is divisible by i (i=2).

The second beautiful arrangement is [2, 1]:

Number at the 1st position (i=1) is 2, and 2 is divisible by i (i=1).

Number at the 2nd position (i=2) is 1, and i (i=2) is divisible by 1.

7. Binary Tree Right Side View

Given 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.

For example: Given the following binary tree,

```
   1            <---
```

```
 /   \
2     3         <---
 \     \
  5     4       <---
```

You should return [1, 3, 4].

8. Car Fleet

N cars are going to the same destination along a one lane road. The destination is target miles away. Each car i has a constant speed speed[i] (in miles per hour), and initial position position[i] miles towards the target along the road. A car can never pass another car ahead of it, but it can catch up to it, and drive bumper to bumper at the same speed. The distance between these two cars is ignored - they are assumed to have the same position. A car fleet is some non-empty set of cars driving at the same position and same speed. Note that a single car is also a car fleet. If a car catches up to a car fleet right at the destination point, it will still be considered as one car fleet. How many car fleets will arrive at the destination?

Example 1:
Input: target = 12, position = [10,8,0,5,3], speed = [2,4,1,1,3]
Output: 3
Explanation:
The cars starting at 10 and 8 become a fleet, meeting each other at 12.
The car starting at 0 doesn’t catch up to any other car, so it is a fleet by itself.
The cars starting at 5 and 3 become a fleet, meeting each other at 6.
Note that no other cars meet these fleets before the destination, so the answer is 3.

9. Coin Change

You are given coins of different denominations and a total amount of money amount. Write a function to compute 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.

Example 1:
coins = [1, 2, 5], amount = 11
return 3 (11 = 5 + 5 + 1)

Example 2:
coins = [2], amount = 3
return -1.

10. Combinations

Given two integers n and k, return all possible combinations of k numbers out of 1 ... n.

For example, If n = 4 and k = 2, a solution is:
[[2,4], [3,4], [2,3], [1,2], [1,3], [1,4],]

11. Combination Sum 3

Find all possible combinations of k numbers that add up to a number n, given that only numbers from 1 to 9 can be used and each combination should be a unique set of numbers.

Example 1:
Input: k = 3, n = 7
Output: [[1,2,4]]

Example 2:
Input: k = 3, n = 9
Output: [[1,2,6], [1,3,5], [2,3,4]]

12. Compare Version

Compare two version numbers version1 and version2. If version1 > version2 return 1, if version1 < version2 return -1, otherwise return 0. You may assume that the version strings are non-empty and contain only digits and the . character. The . character does not represent a decimal point and is used to separate number sequences. For instance, 2.5 is not "two and a half" or "half way to version three", it is the fifth second-level revision of the second first-level revision.
Here is an example of version numbers ordering:
0.1 < 1.1 < 1.2 < 13.37

13. Construct Binary Tree From String

You need to construct a binary tree from a string consisting of parenthesis and integers. The whole input represents a binary tree. It contains an integer followed by zero, one or two pairs of parenthesis. The integer represents the root's value and a pair of parenthesis contains a child binary tree with the same structure. You always start to construct the left child node of the parent first if it exists.
Example:
Input: "4(2(3)(1))(6(5))"
Output: return the tree root node representing the following tree:

```
       4
     /   \
    2     6
   / \   /
  3   1 5
```

Note:
There will only be '(', ')', '-' and '0' ~ '9' in the input string. An empty tree is represented by "" instead of "()".

14. Container With Most Water

Given n non-negative integers a1, a2, ..., an, where each represents a point at coordinate (i, ai). n vertical lines are drawn such that the two endpoints of line i is at (i, ai) and (i, 0). Find two lines, which together with x-axis forms a container, such that the container contains the most water.
Note: You may not slant the container and n is at least 2.

15. Continuous Subarray Sum

Given a list of non-negative numbers and a target integer k, write a function to check if the array has a continuous subarray of size at least 2 that sums up to the multiple of k, that is, sums up to n*k where n is also an integer.

Example 1:
Input: [23, 2, 4, 6, 7], k=6 Output: True
Explanation: Because [2, 4] is a continuous subarray of size 2 and sums up to 6.
Example 2:
Input: [23, 2, 6, 4, 7], k=6 Output: True
Explanation: Because [23, 2, 6, 4, 7] is an continuous subarray of size 5 and sums up to 42.

16. Custom Sort String

S and T are strings composed of lowercase letters. In S, no letter occurs more than once.
S was sorted in some custom order previously. We want to permute the characters of T so that they match the order that S was sorted. More specifically, if x occurs before y in S, then x should occur before y in the returned string.
Return any permutation of T (as a string) that satisfies this property.
Example :
Input:
S = "cba"
T = "abcd"
Output: "cbad"
Explanation:
"a", "b", "c" appear in S, so the order of "a", "b", "c" should be "c", "b", and "a".
Since "d" does not appear in S, it can be at any position in T. "dcba", "cdba", "cbda" are also valid outputs.
Note:
S has length at most 26, and no character is repeated in S.
T has length at most 200.
S and T consist of lowercase letters only.

17. Course Schedule

There are a total of n courses you have to take, labeled from 0 to n - 1. Some courses may have prerequisites, for example to take course 0 you have to first take course 1, which is expressed as a pair: [0,1]. Given the total number of courses and a list of prerequisite pairs, is it possible for you to finish all courses?
For example:
Example 1: 2, [[1,0]]
There are a total of 2 courses to take. To take course 1 you should have finished course 0. So it is possible.
Example 2: 2, [[1,0],[0,1]]
There are a total of 2 courses to take. To take course 1 you should have finished course 0, and to take course 0 you should also have finished course 1. So it is impossible.
Note:
The input prerequisites is a graph represented by a list of edges, not adjacency matrices. Read more about how a graph is represented.

18. Course Schedule II

There are a total of n courses you have to take, labeled from 0 to n - 1. Some courses may have prerequisites, for example to take course 0 you have to first take course 1, which is expressed as a pair: [0,1]. Given the total number of courses and a list of prerequisite pairs, return the ordering of courses you should take to finish all courses. There may be multiple correct orders, you just need to return one of them. If it is impossible to finish all courses, return an empty array.

For example:

Example 1: 2, [[1,0]]

There are a total of 2 courses to take. To take course 1 you should have finished course 0. So the correct course order is [0,1]

Example 2: 4, [[1,0],[2,0],[3,1],[3,2]]

There are a total of 4 courses to take. To take course 3 you should have finished both courses 1 and 2. Both courses 1 and 2 should be taken after you finished course 0. So one correct course order is [0,1,2,3]. Another correct ordering is[0,2,1,3].

Note:

The input prerequisites is a graph represented by a list of edges, not adjacency matrices. Read more about how a graph is represented.

19. Cut Off Trees for Golf Event

You are asked to cut off trees in a forest for a golf event. The forest is represented as a non-negative 2D map, in this map:

0 represents the obstacle can't be reached.

1 represents the ground can be walked through.

The place with number bigger than 1 represents a tree can be walked through, and this positive number represents the tree's height.

You are asked to cut off all the trees in this forest in the order of tree's height - always cut off the tree with lowest height first. And after cutting, the original place has the tree will become a grass (value 1).

You will start from the point (0, 0) and you should output the minimum steps you need to walk to cut off all the trees. If you can't cut off all the trees, output -1 in that situation.

You are guaranteed that no two trees have the same height and there is at least one tree needs to be cut off.

Example 1:

```
Input:
[
 [1,2,3],
 [0,0,4],
 [7,6,5]
]
Output: 6
```

Example 2:

```
Input:
[
 [1,2,3],
 [0,0,0],
 [7,6,5]
]
```

Output: -1
Example 3:
Input:
```
[
 [2,3,4],
 [0,0,5],
 [8,7,6]
]
```
Output: 6
Explanation: You started from the point (0,0) and you can cut off the tree in (0,0) directly without walking.
Hint: size of the given matrix will not exceed 50x50.

20. Daily Temperature

Given a list of daily temperatures, produce a list that, for each day in the input, tells you how many days you would have to wait until a warmer temperature. If there is no future day for which this is possible, put 0 instead.
For example, given the list temperatures = [73, 74, 75, 71, 69, 72, 76, 73], your output should be [1, 1, 4, 2, 1, 1, 0, 0].

21. Decode String

Given an encoded string, return it's decoded string. The encoding rule is: k[encoded_string], where the encoded_string inside the square brackets is being repeated exactly k times. Note that k is guaranteed to be a positive integer. You may assume that the input string is always valid; No extra white spaces, square brackets are well-formed, etc. Furthermore, you may assume that the original data does not contain any digits and that digits are only for those repeat numbers, k. For example, there won't be input like 3a or 2[4].
Examples:
s = ″3[a]2[bc]″, return ″aaabcbc″.
s = ″3[a2[c]]″, return ″accaccacc″.
s = ″2[abc]3[cd]ef″, return ″abcabccdcdcdef″.

22. Delete Node in a BST

Given a root node reference of a BST and a key, delete the node with the given key in the BST. Return the root node reference (possibly updated) of the BST.

23. Employee Importance

You are given a data structure of employee information, which includes the employee's unique id, his importance value and his direct subordinates' id.
For example, employee 1 is the leader of employee 2, and employee 2 is the leader of employee 3. They have importance value 15, 10 and 5, respectively. Then employee 1 has a data structure like [1, 15, [2]], and employee 2 has [2, 10, [3]], and employee 3 has [3, 5, []]. Note that although employee 3 is also a subordinate of employee 1, the relationship is not direct.
Now given the employee information of a company, and an employee id, you need to return the total importance value of this employee and all his subordinates.
Example 1:

Input: [[1, 5, [2, 3]], [2, 3, []], [3, 3, []]], 1
Output: 11
Explanation:
Employee 1 has importance value 5, and he has two direct subordinates: employee 2 and employee 3. They both have importance value 3. So the total importance value of employee 1 is 5 + 3 + 3 = 11.
Note:
One employee has at most one direct leader and may have several subordinates.
The maximum number of employees won't exceed 2000.

24. Equal Tree Partition

Given a binary tree with n nodes, your task is to check if it's possible to partition the tree to two trees which have the equal sum of values after removing exactly one edge on the original tree.

Example 1:
Input:

```
    5
   / \
  10 10
    /  \
   2   3
```

Output: True
Explanation:

```
    5
   /
  10
```

Sum: 15

```
   10
  /  \
 2    3
```

Sum: 15
Example 2:
Input:

```
    1
   / \
  2  10
    /  \
   2   20
```

Output: False
Explanation: You can't split the tree into two trees with equal sum after removing exactly one edge on the tree.

25. Find And Replace in String

To some string S, we will perform some replacement operations that replace groups of letters with new ones (not necessarily the same size). Each replacement operation has 3 parameters: a starting index i, a source word x and a target word y. The rule is that if x starts at position i in the original string S, then we will replace that occurrence of x with y. If not, we do nothing.

For example, if we have S = "abcd" and we have some replacement operation i = 2, x = "cd", y = "ffff", then because "cd" starts at position 2 in the original string S, we will replace it with "ffff".

Using another example on S = "abcd", if we have both the replacement operation i = 0, x = "ab", y = "eee", as well as another replacement operation i = 2, x = "ec", y = "ffff", this second operation does nothing because in the original string S[2] = 'c', which doesn't match x[0] = 'e'.

All these operations occur simultaneously. It's guaranteed that there won't be any overlap in replacement: for example, S = "abc", indexes = [0, 1], sources = ["ab","bc"] is not a valid test case.

Example 1:
Input: S = "abcd", indexes = [0,2], sources = ["a","cd"], targets = ["eee","ffff"]
Output: "eeebffff"
Explanation: "a" starts at index 0 in S, so it's replaced by "eee". "cd" starts at index 2 in S, so it's replaced by "ffff".

Example 2:
Input: S = "abcd", indexes = [0,2], sources = ["ab","ec"], targets = ["eee","ffff"]
Output: "eeecd"
Explanation: "ab" starts at index 0 in S, so it's replaced by "eee". "ec" doesn't starts at index 2 in the original S, so we do nothing.

26. Find Duplicate File in System

Given a list of directory info including directory path, and all the files with contents in this directory, you need to find out all the groups of duplicate files in the file system in terms of their paths. A group of duplicate files consists of at least two files that have exactly the same content.

A single directory info string in the input list has the following format:
"root/d1/d2/.../dm f1.txt(f1_content) f2.txt(f2_content) ... fn.txt(fn_content)"
It means there are n files (f1.txt, f2.txt ... fn.txt with content f1_content, f2_content ... fn_content, respectively) in directory root/d1/d2/.../dm. Note that n >= 1 and m >= 0. If m = 0, it means the directory is just the root directory.

The output is a list of group of duplicate file paths. For each group, it contains all the file paths of the files that have the same content. A file path is a string that has the following format:
"directory_path/file_name.txt"

Example 1:
Input: ["root/a 1.txt(abcd) 2.txt(efgh)", "root/c 3.txt(abcd)", "root/c/d 4.txt(efgh)",

"root 4.txt(efgh)"]

Output: [["root/a/2.txt","root/c/d/4.txt","root/4.txt"],["root/a/1.txt","root/c/3.txt"]]

Note:

No order is required for the final output.

You may assume the directory name, file name and file content only has letters and digits, and the length of file content is in the range of [1,50].

The number of files given is in the range of [1,20000].

You may assume no files or directories share the same name in a same directory.

You may assume each given directory info represents a unique directory. Directory path and file infos are separated by a single blank space.

27. Find Largest Element in Each Row

You need to find the largest value in each row of a binary tree.

Example:

Input:

```
          1
         / \
        3   2
       / \   \
      5   3   9
```

Output: [1, 3, 9]

28. Find Peak Element

A peak element is an element that is greater than its neighbors.

Given an input array where num[i] ≠ num[i+1], find a peak element and return its index.

The array may contain multiple peaks, in that case return the index to any one of the peaks is fine.

You may imagine that num[-1] = num[n] = -∞.

For example, in array [1, 2, 3, 1], 3 is a peak element and your function should return the index number 2.

29. Four Sum II

Given four lists A, B, C, D of integer values, compute how many tuples (i, j, k, l) there are such that A[i] + B[j] + C[k] + D[l] is zero. To make problem a bit easier, all A, B, C, D have same length of N where 0 ≤ N ≤ 500. All integers are in the range of -228 to 228 - 1 and the result is guaranteed to be at most 231 - 1.

Example:

Input:

A = [1, 2]

B = [-2,-1]

C = [-1, 2]

D = [0, 2]

Output: 2

Explanation:

The two tuples are:

1. (0, 0, 0, 1) -> A[0] + B[0] + C[0] + D[1] = 1 + (-2) + (-1) + 2 = 0

2. (1, 1, 0, 0) -> A[1] + B[1] + C[0] + D[0] = 2 + (-1) + (-1) + 0 = 0

30. Friend Circles

There are N students in a class. Some of them are friends, while some are not. Their friendship is transitive in nature. For example, if A is a direct friend of B, and B is a direct friend of C, then A is an indirect friend of C. And we defined a friend circle is a group of students who are direct or indirect friends.

Given a N*N matrix M representing the friend relationship between students in the class. If M[i][j] = 1, then the ith and jth students are direct friends with each other, otherwise not. And you have to output the total number of friend circles among all the students.

Example 1:

```
Input:
[[1,1,0],
 [1,1,0],
 [0,0,1]]
Output: 2
```

Explanation:The 0th and 1st students are direct friends, so they are in a friend circle. The 2nd student himself is in a friend circle. So return 2.

Example 2:

```
Input:
[[1,1,0],
 [1,1,1],
 [0,1,1]]
Output: 1
```

Explanation:The 0th and 1st students are direct friends, the 1st and 2nd students are direct friends,

so the 0th and 2nd students are indirect friends. All of them are in the same friend circle, so return 1.

Note:

N is in range [1,200].

M[i][i] = 1 for all students.

If M[i][j] = 1, then M[j][i] = 1.

31. Game of Life

Given a board with m by n cells, each cell has an initial state live (1) or dead (0).

Each cell interacts with its eight neighbors (horizontal, vertical, diagonal) using the following four rules (taken from the above Wikipedia article):

Any live cell with fewer than two live neighbors dies, as if caused by under-population.

Any live cell with two or three live neighbors lives on to the next generation.

Any live cell with more than three live neighbors dies, as if by over-population..

Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

Write a function to compute the next state (after one update) of the board given its current state.

32. Increasing Subsequences

Given an integer array, your task is to find all the different possible increasing subsequences of the given array, and the length of an increasing subsequence should be at least 2 .

Example:
Input: [4, 6, 7, 7]
Output: [[4, 6], [4, 7], [4, 6, 7], [4, 6, 7, 7], [6, 7], [6, 7, 7], [7,7], [4,7,7]]
Note:
The length of the given array will not exceed 15.
The range of integer in the given array is [-100,100].
The given array may contain duplicates, and two equal integers should also be considered as a special case of increasing sequence.

33. Is Graph Bipartite?

Given a graph, return true if and only if it is bipartite. Recall that a graph is bipartite if we can split it's set of nodes into two independent subsets A and B such that every edge in the graph has one node in A and another node in B. The graph is given in the following form: graph[i] is a list of indexes [j,k,…] for which the edges between nodes i and j, i and k, i and … exist. Each node is an integer between 0 and graph.length - 1. There are no self edges or parallel edges: graph[i] does not contain i, and it doesn't contain any element twice.

Example 1:
Input: [[1,3], [0,2], [1,3], [0,2]]
Output: true
Explanation:
The graph looks like this:

```
0----1
|    |
|    |
3----2
```

We can divide the vertices into two groups: {0, 2} and {1, 3}.
Example 2:
Input: [[1,2,3], [0,2], [0,1,3], [0,2]]
Output: false
Explanation:
The graph looks like this:

```
0----1
| \  |
|  \ |
3----2
```

We cannot find a way to divide the set of nodes into two independent subsets.

34. Jump Game

Given an array of non-negative integers, you are initially positioned at the first index of the array. Each element in the array represents your maximum jump length at that position. Determine if you are able to reach the last index.
For example:
A = [2,3,1,1,4], return true.
A = [3,2,1,0,4], return false.

35. Kth Largest Element in an Array

Find the kth largest element in an unsorted array.

For example,Given [3,2,1,5,6,4] and k = 2, return 5.

36. License Key Formatting

Now you are given a string S, which represents a software license key which we would like to format. The string S is composed of alphanumerical characters and dashes. The dashes split the alphanumerical characters within the string into groups. (i.e. if there are M dashes, the string is split into M+1 groups). The dashes in the given string are possibly misplaced.

We want each group of characters to be of length K (except for possibly the first group, which could be shorter, but still must contain at least one character). To satisfy this requirement, we will reinsert dashes. Additionally, all the lower case letters in the string must be converted to upper case.

So, you are given a non-empty string S, representing a license key to format, and an integer K. And you need to return the license key formatted according to the description above.

Example 1:

Input: S = "2-4A0r7-4k", K = 4

Output: "24A0-R74K"

Explanation: The string S has been split into two parts, each part has 4 characters.

37. Linked List Components

We are given head, the head node of a linked list containing unique integer values.

We are also given the list G, a subset of the values in the linked list.

Return the number of connected components in G, where two values are connected if they appear consecutively in the linked list.

Example 1:

Input:

head: 0->1->2->3

G = [0, 1, 3]

Output: 2

Explanation:

0 and 1 are connected, so [0, 1] and [3] are the two connected components.

Example 2:

Input:

head: 0->1->2->3->4

G = [0, 3, 1, 4]

Output: 2

Explanation:

0 and 1 are connected, 3 and 4 are connected, so [0, 1] and [3, 4] are the two connected components.

Note:

If N is the length of the linked list given by head, 1 <= N <= 10000.

The value of each node in the linked list will be in the range [0, N - 1].

1 <= G.length <= 10000.

G is a subset of all values in the linked list.

38. Longest Palindromic Substring

Given a string s, find the longest palindromic substring in s. You may assume that the

maximum length of s is 1000.
Example 1:
Input: ″babad″
Output: ″bab″
Note: ″aba″ is also a valid answer.
Example 2:
Input: ″cbbd″
Output: ″bb″

39. Longest Univalue Path

Given a binary tree, find the length of the longest path where each node in the path has the same value. This path may or may not pass through the root.
Note: The length of path between two nodes is represented by the number of edges between them.

Example 1:
Input:

```
              5
             / \
            4   5
           / \   \
          1   1   5
```

Output:2

Example 2:
Input:

```
              1
             / \
            4   5
           / \   \
          4   4   5
```

Output:2
Note: The given binary tree has not more than 10000 nodes. The height of the tree is not more than 1000.

40. Longest Word in Dictionary through Deleting

Given a string and a string dictionary, find the longest string in the dictionary that can be formed by deleting some characters of the given string. If there are more than one possible results, return the longest word with the smallest lexicographical order. If there is no possible result, return the empty string.

Example 1:
Input: s = ″abpcplea″, d = [″ale″,″apple″,″monkey″,″plea″]
Output: ″apple″
Example 2:
Input: s = ″abpcplea″, d = [″a″,″b″,″c″]
Output: ″a″

41. Lowest Common Ancestor of Binary Tree

Given a binary tree, find the lowest common ancestor (LCA) of two given nodes in the tree. According to the definition of LCA on Wikipedia: “The lowest common ancestor is defined between two nodes v and w as the lowest node in T that has both v and w as descendants (where we allow a node to be a descendant of itself).”

```
        _______3______
       /              \
    ___5__          ___1__
   /      \        /      \
   6      _2       0       8
         /  \
         7   4
```

For example, the lowest common ancestor (LCA) of nodes 5 and 1 is 3. Another example is LCA of nodes 5 and 4 is 5, since a node can be a descendant of itself according to the LCA definition

42. Map Sum

Implement a MapSum class with insert, and sum methods. For the method insert, you'll be given a pair of (string, integer). The string represents the key and the integer represents the value. If the key already existed, then the original key-value pair will be overridden to the new one. For the method sum, you'll be given a string representing the prefix, and you need to return the sum of all the pairs' value whose key starts with the prefix.

Example 1:
Input: insert(″apple″, 3), Output: Null
Input: sum(″ap″), Output: 3
Input: insert(″app″, 2), Output: Null
Input: sum(″ap″), Output: 5

43. Maximum Length of Repeated Subarray

Given two integer arrays A and B, return the maximum length of an subarray that appears in both arrays.

Example 1:
Input: A: [1,2,3,2,1]   B: [3,2,1,4,7]
Output: 3
Explanation: The repeated subarray with maximum length is [3, 2, 1].

44. Maximum Width of Binary Tree

Given a binary tree, write a function to get the maximum width of the given tree. The width of a tree is the maximum width among all levels. The binary tree has the same structure as a full binary tree, but some nodes are null. The width of one level is defined as the length between the end-nodes (the leftmost and right most non-null nodes in the level, where the null nodes between the end-nodes are also counted into the length calculation.

Example 1:
Input:

```
           1
         /   \
        3     2
       / \     \
      5   3     9
```

Output: 4
Explanation: The maximum width existing in the third level with the length 4 (5,3,null,9).
Example 2:
Input:

```
          1
         /
        3
       / \
      5   3
```

Output: 2
Explanation: The maximum width existing in the third level with the length 2 (5,3).
Example 3:
Input:

```
          1
         / \
        3   2
       /
      5
```

Output: 2
Explanation: The maximum width existing in the second level with the length 2 (3,2).
Example 4:
Input:

```
          1
         / \
        3   2
       /     \
      5       9
     /         \
    6           7
```

Output: 8
Explanation: The maximum width existing in the fourth level with the length 8 (6,null,null,null,null,null,null,7).

45. Merge k Sorted Lists

There are k sorted lists. Merge these k sorted lists into one sorted lists.

46. Minimum Height Trees

For a undirected graph with tree characteristics, we can choose any node as the root. The result graph is then a rooted tree. Among all possible rooted trees, those with minimum height are called minimum height trees (MHTs). Given such a graph, write a function to find all the MHTs and return a list of their root labels.
The graph contains n nodes which are labeled from 0 to n - 1. You will be given the number n and a list of undirected edges (each edge is a pair of labels). You can assume that no duplicate edges will appear in edges. Since all edges are undirected, [0, 1] is the same as [1, 0] and thus will not appear together in edges.
Example 1:
Given n = 4, edges = [[1, 0], [1, 2], [1, 3]]

```
        0
        |
        1
       / \
      2   3
```

return [1]

47. Minimum Index Sum of Two Lists

Suppose Andy and Doris want to choose a restaurant for dinner, and they both have a list of favorite restaurants represented by strings. You need to help them find out their common interest with the least list index sum. If there is a choice tie between answers, output all of them with no order requirement. You could assume there always exists an answer.

Example 1:
Input:
["Shogun", "Tapioca Express", "Burger King", "KFC"]
["Piatti", "The Grill at Torrey Pines", "Hungry Hunter Steakhouse", "Shogun"]
Output: ["Shogun"]
Explanation: The only restaurant they both like is "Shogun".
Example 2:
Input:
["Shogun", "Tapioca Express", "Burger King", "KFC"]
["KFC", "Shogun", "Burger King"]
Output: ["Shogun"]
Explanation: The restaurant they both like and have the least index sum is "Shogun" with index sum 1 (0+1).

48. Monotone Increasing Digits

Given a non-negative integer N, find the largest number that is less than or equal to N with monotone increasing digits. Recall that an integer has monotone increasing digits if and only if each pair of adjacent digits x and y satisfy x <= y.

Example 1:
Input: N = 10

Output: 9
Example 2:
Input: N = 1234
Output: 1234
Example 3:
Input: N = 332
Output: 299

49. Multiply Strings
Given two non-negative integers num1 and num2 represented as strings, return the product of num1 and num2.

Note:
The length of both num1 and num2 is < 110.
Both num1 and num2 contains only digits 0-9.
Both num1 and num2 does not contain any leading zero.

50. My Calendar I
Implement a MyCalendar class to store your events. A new event can be added if adding the event will not cause a double booking. Your class will have the method, book(int start, int end). Formally, this represents a booking on the half open interval [start, end), the range of real numbers x such that start <= x < end. A double booking happens when two events have some non-empty intersection (ie., there is some time that is common to both events.) For each call to the method MyCalendar.book, return true if the event can be added to the calendar successfully without causing a double booking. Otherwise, return false and do not add the event to the calendar.
Your class will be called like this: MyCalendar cal = new MyCalendar();
MyCalendar.book(start, end)
Example 1:
MyCalendar();
MyCalendar.book(10, 20); // returns true
MyCalendar.book(15, 25); // returns false
MyCalendar.book(20, 30); // returns true
Explanation:
The first event can be booked. The second can't because time 15 is already booked by another event. The third event can be booked, as the first event takes every time less than 20, but not including 20.

51. Network Delay
There are N network nodes, labelled 1 to N. Given times, a list of travel times as directed edges times[i] = (u, v, w), where u is the source node, v is the target node, and w is the time it takes for a signal to travel from source to target. Now, we send a signal from a certain node K. How long will it take for all nodes to receive the signal? If it is impossible, return -1.

Note:
N will be in the range [1, 100].
K will be in the range [1, N].
The length of times will be in the range [1, 6000].

All edges times[i] = (u, v, w) will have 1 <= u, v <= N and 1 <= w <= 100.

52. Next Closest Time

Given a time represented in the format "HH:MM", form the next closest time by reusing the current digits. There is no limit on how many times a digit can be reused. You may assume the given input string is always valid. For example, "01:34", "12:09" are all valid. "1:34", "12:9" are all invalid.

Example 1:
Input: "19:34"
Output: "19:39"
Explanation: The next closest time choosing from digits 1, 9, 3, 4, is 19:39, which occurs 5 minutes later. It is not 19:33, because this occurs 23 hours and 59 minutes later.

Example 2:
Input: "23:59"
Output: "22:22"
Explanation: The next closest time choosing from digits 2, 3, 5, 9, is 22:22. It may be assumed that the returned time is next day's time since it is smaller than the input time numerically.

53. Next Permutation

Implement next permutation, which rearranges numbers into the lexicographically next greater permutation of numbers. If such arrangement is not possible, it must rearrange it as the lowest possible order (ie, sorted in ascending order). The replacement must be in-place, do not allocate extra memory.
Here are some examples. Inputs are in the left-hand column and its corresponding outputs are in the right-hand column.
1,2,3 → 1,3,2
3,2,1 → 1,2,3
1,1,5 → 1,5,1

54. Number of Matching Subsequences

Given string S and a dictionary of words words, find the number of words[i] that is a subsequence of S.

Example :
Input: S = "abcde"
words = ["a", "bb", "acd", "ace"]
Output: 3
Explanation: There are three words in words that are a subsequence of S: "a", "acd", "ace".

55. Palindrome Permutation 2

Given a string s, return all the palindromic permutations (without duplicates) of it. Return an empty list if no palindromic permutation could be form.
For example:
Given s = "aabb", return ["abba", "baab"].

Given s = ″abc″, return [].

56. Possible Bipartition

Given a set of N people (numbered 1, 2, ..., N), we would like to split everyone into two groups of any size. Each person may dislike some other people, and they should not go into the same group. Formally, if dislikes[i] = [a, b], it means it is not allowed to put the people numbered a and b into the same group. Return true if and only if it is possible to split everyone into two groups in this way.

Example 1:
Input: N = 4, dislikes = [[1,2],[1,3],[2,4]]
Output: true
Explanation: group1 [1,4], group2 [2,3]
Example 2:
Input: N = 5, dislikes = [[1,2],[2,3],[3,4],[4,5],[1,5]]
Output: false

Note:
dislikes[i][0] < dislikes[i][1]
There does not exist i != j for which dislikes[i] == dislikes[j].

57. Queue Reconstruction by Height

Suppose you have a random list of people standing in a queue. Each person is described by a pair of integers (h, k), where h is the height of the person and k is the number of people in front of this person who have a height greater than or equal to h. Write an algorithm to reconstruct the queue.
Example
Input: [[7,0], [4,4], [7,1], [5,0], [6,1], [5,2]]
Output: [[5,0], [7,0], [5,2], [6,1], [4,4], [7,1]]

58. Remove Comments

Given a C++ program, remove comments from it. The program source is an array where source[i] is the i-th line of the source code. This represents the result of splitting the original source code string by the newline character \n.
In C++, there are two types of comments, line comments, and block comments.
The string // denotes a line comment, which represents that it and rest of the characters to the right of it in the same line should be ignored.
The string /* denotes a block comment, which represents that all characters until the next (non-overlapping) occurrence of */ should be ignored. (Here, occurrences happen in reading order: line by line from left to right.) To be clear, the string /*/ does not yet end the block comment, as the ending would be overlapping the beginning.
The first effective comment takes precedence over others: if the string // occurs in a block comment, it is ignored. Similarly, if the string /* occurs in a line or block comment, it is also ignored. If a certain line of code is empty after removing comments, you must not output that line: each string in the answer list will be non-empty.
There will be no control characters, single quote, or double quote characters. For example, source = ″string s = ″/* Not a comment. */″;″ will not be a test case. (Also, nothing else such as defines or macros will interfere with the comments.)
It is guaranteed that every open block comment will eventually be closed, so /* outside

of a line or block comment always starts a new comment.
Finally, implicit newline characters can be deleted by block comments. Please see the examples below for details.
After removing the comments from the source code, return the source code in the same format.

Example 1:
Input:
source = ["/*Test program */", "int main()", "{ ", " // variable declaration ", "int a, b, c;", "/* This is a test", "   multiline ", "   comment for ", "   testing */", "a = b + c;", "}"]

The line by line code is visualized as below:

```
/*Test program */
int main()
{
  // variable declaration
int a, b, c;
/* This is a test
   multiline
   comment for
   testing */
a = b + c;
}
```

Output: ["int main()","{ "," ","int a, b, c;","a = b + c;","}"]

The line by line code is visualized as below:

```
int main()
{
int a, b, c;
a = b + c;
}
```

Explanation:
The string
/*
 denotes a block comment, including line 1 and lines 6-9. The string
//
 denotes line 4 as comments.
Example 2:
Input:
source = ["a/*comment", "line", "more_comment*/b"]
Output: ["ab"]
Explanation: The original source string is "a/*comment\nline\nmore_comment*/b", where we have bolded the newline characters. After deletion, the implicit newline characters are deleted, leaving the string "ab", which when delimited by newline characters becomes ["ab"].
Note:

The length of source is in the range [1, 100].
The length of source[i] is in the range [0, 80].
Every open block comment is eventually closed.
There are no single-quote, double-quote, or control characters in the source code.

59. Replace Words

In English, we have a concept called root, which can be followed by some other words to form another longer word - let's call this word successor. For example, the root an, followed by other, which can form another word another. Now, given a dictionary consisting of many roots and a sentence. You need to replace all the successor in the sentence with the root forming it. If a successor has many roots can form it, replace it with the root with the shortest length. You need to output the sentence after the replacement.

Example 1:
Input: dict = ["cat", "bat", "rat"]
sentence = "the cattle was rattled by the battery"
Output: "the cat was rat by the bat"
Note:
The input will only have lower-case letters.
1 <= dict words number <= 1000
1 <= sentence words number <= 1000
1 <= root length <= 100
1 <= sentence words length <= 1000

60. Reverse Linked List II

Reverse a linked list from position m to n. Do it in-place and in one-pass.
For example:
Given 1->2->3->4->5->NULL, m = 2 and n = 4,
return 1->4->3->2->5->NULL.
Note:
Given m, n satisfy the following condition:
1 ≤ m ≤ n ≤ length of list.

61. Search in Rotated Sorted Array

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand. (i.e., 0 1 2 4 5 6 7 might become 4 5 6 7 0 1 2). You are given a target value to search. If found in the array return its index, otherwise return -1. You may assume no duplicate exists in the array.

62. Search in Rotated Sorted Array 2

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand.
(i.e., 0 1 2 4 5 6 7 might become 4 5 6 7 0 1 2). Write a function to determine if a given target is in the array. The array may contain duplicates.

63. Serialize and Deserialize BST

Serialization is the process of converting a data structure or object into a sequence of bits so that it can be stored in a file or memory buffer, or transmitted across a network connection link to be reconstructed later in the same or another computer environment.

Design an algorithm to serialize and deserialize a binary search tree. There is no restriction on how your serialization/deserialization algorithm should work. You just need to ensure that a binary search tree can be serialized to a string and this string can be deserialized to the original tree structure.
The encoded string should be as compact as possible.

64. Set Matrix Zeroes
Given a m*n matrix, if an element is 0, set its entire row and column to 0. Do it in place.

65. Shopping Offers
In LeetCode Store, there are some kinds of items to sell. Each item has a price. However, there are some special offers, and a special offer consists of one or more different kinds of items with a sale price. You are given the each item's price, a set of special offers, and the number we need to buy for each item. The job is to output the lowest price you have to pay for exactly certain items as given, where you could make optimal use of the special offers. Each special offer is represented in the form of an array, the last number represents the price you need to pay for this special offer, other numbers represents how many specific items you could get if you buy this offer. You could use any of special offers as many times as you want.
Example 1:
Input: [2,5], [[3,0,5],[1,2,10]], [3,2]
Output: 14
Explanation: There are two kinds of items, A and B. Their prices are $2 and $5 respectively.
In special offer 1, you can pay $5 for 3A and 0B
In special offer 2, you can pay $10 for 1A and 2B.
You need to buy 3A and 2B, so you may pay $10 for 1A and 2B (special offer #2), and $4 for 2A.
Example 2:
Input: [2,3,4], [[1,1,0,4],[2,2,1,9]], [1,2,1]
Output: 11
Explanation: The price of A is $2, and $3 for B, $4 for C.
You may pay $4 for 1A and 1B, and $9 for 2A ,2B and 1C.
You need to buy 1A ,2B and 1C, so you may pay $4 for 1A and 1B (special offer #1), and $3 for 1B, $4 for 1C.
You cannot add more items, though only $9 for 2A ,2B and 1C.
Note:
There are at most 6 kinds of items, 100 special offers.
For each item, you need to buy at most 6 of them.
You are not allowed to buy more items than you want, even if that would lower the overall price.

66. Short Encoding of Words
Given a list of words, we may encode it by writing a reference string S and a list of indexes A. For example, if the list of words is ["time", "me", "bell"], we can write it

as S = ″time#bell#″ and indexes = [0, 2, 5]. Then for each index, we will recover the word by reading from the reference string from that index until we reach a ″#″ character. What is the length of the shortest reference string S possible that encodes the given words?
Example:
Input: words = [″time″, ″me″, ″bell″]
Output: 10
Explanation: S = ″time#bell#″ and indexes = [0, 2, 5].

67. Smallest Subtree with all the Deepest Nodes
Given a binary tree rooted at root, the depth of each node is the shortest distance to the root. A node is deepest if it has the largest depth possible among any node in the entire tree. Return the node with the largest depth such that it contains all the deepest nodes in its subtree.
Example 1:
Input: [3,5,1,6,2,0,8,null,null,7,4]
Output: [2,7,4]
Explanation:
We return the node with value 2, colored in yellow in the diagram.
The nodes colored in blue are the deepest nodes of the tree.
The input ″[3, 5, 1, 6, 2, 0, 8, null, null, 7, 4]″ is a serialization of the given tree.
The output ″[2, 7, 4]″ is a serialization of the subtree rooted at the node with value 2.
Both the input and output have TreeNode type.

68. Sort List
Sort a linked list in O(n log n) time using constant space complexity.

69. Split Linked List in Parts
Given a (singly) linked list with head node root, write a function to split the linked list into k consecutive linked list ″parts″. The length of each part should be as equal as possible: no two parts should have a size differing by more than 1. This may lead to some parts being null. The parts should be in order of occurrence in the input list, and parts occurring earlier should always have a size greater than or equal parts occurring later. Return a List of ListNode's representing the linked list parts that are formed.
Examples 1->2->3->4, k = 5 // 5 equal parts [[1], [2], [3], [4], null]
Example 1:
Input: root = [1, 2, 3], k = 5
Output: [[1],[2],[3],[],[]]
Explanation: The input and each element of the output are ListNodes, not arrays.
Example 2:
Input: root = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], k = 3
Output: [[1, 2, 3, 4], [5, 6, 7], [8, 9, 10]]
Explanation:
The input has been split into consecutive parts with size difference at most 1, and earlier parts are a larger size than the later parts.

70. Subarray Product Less Than K

Your are given an array of positive integers nums.Count and print the number of (contiguous) subarrays where the product of all the elements in the subarray is less than k.

Example 1:

Input: nums = [10, 5, 2, 6], k = 100

Output: 8

Explanation: The 8 subarrays that have product less than 100 are: [10], [5], [2], [6], [10, 5], [5, 2], [2, 6], [5, 2, 6].

Note that [10, 5, 2] is not included as the product of 100 is not strictly less than k.

71. Subsets II

Given a collection of integers that might contain duplicates, nums, return all possible subsets.

Note: The solution set must not contain duplicate subsets.

For example,

If nums = [1,2,2], a solution is:

```
[
  [2],
  [1],
  [1,2,2],
  [2,2],
  [1,2],
  []
]
```

72. Sum 3

Given an array S of n integers, are there elements a, b, c in S such that a+b+c=0? Find all unique triplets in the array which gives the sum of zero.

Note: The solution set must not contain duplicate triplets.

For example, given array S = [-1, 0, 1, 2, -1, -4],

A solution set is:

```
[
  [-1, 0, 1],
  [-1, -1, 2]
]
```

73. Task Scheduler

Given a char array representing tasks CPU need to do. It contains capital letters A to Z where different letters represent different tasks.Tasks could be done without original order. Each task could be done in one interval. For each interval, CPU could finish one task or just be idle. However, there is a non-negative cooling interval n that means between two same tasks, there must be at least n intervals that CPU are doing different tasks or just be idle. You need to return the least number of intervals the CPU will take to finish all the given tasks.

Example 1:

Input: tasks = ['A','A','A','B','B','B'], n = 2

Output: 8

Explanation: A -> B -> idle -> A -> B -> idle -> A -> B.

74. Top K Frequent Elements
Given a non-empty array of integers, return the k most frequent elements.
For example,
Given [1,1,1,2,2,3] and k = 2, return [1,2].
Note:
You may assume k is always valid, 1 ≤ k ≤ number of unique elements.
Your algorithm's time complexity must be better than O(n log n), where n is the array's size.

75. Triangle
Given a triangle, find the minimum path sum from top to bottom. Each step you may move to adjacent numbers on the row below.
For example, given the following triangle

```
[
     [2],
    [3,4],
   [6,5,7],
  [4,1,8,3]
]
```

The minimum path sum from top to bottom is 11 (i.e., 2 + 3 + 5 + 1 = 11).

76. Two Keys Keyboard
Initially on a notepad only one character 'A' is present. You can perform two operations on this notepad for each step:
Copy All: You can copy all the characters present on the notepad (partial copy is not allowed).
Paste: You can paste the characters which are copied last time.
Given a number n. You have to get exactly n 'A' on the notepad by performing the minimum number of steps permitted. Output the minimum number of steps to get n 'A'.
Example 1:
Input: 3
Output: 3
Explanation:
Intitally, we have one character 'A'.
In step 1, we use Copy All operation.
In step 2, we use Paste operation to get 'AA'.
In step 3, we use Paste operation to get 'AAA'.

77. Unique Paths II
A robot is located at the top-left corner of a m*n grid. The robot can only move either down or right at any point in time. The robot is trying to reach the bottom-right corner of the grid. There are some obstacles in the grids. How many unique paths would there be?
An obstacle and empty space is marked as 1 and 0 respectively in the grid.
For example, There is one obstacle in the middle of a 3x3 grid as illustrated below.

```
[
  [0,0,0],
  [0,1,0],
  [0,0,0]
]
```
The total number of unique paths is 2.

78. Validate Binary Search Tree

Given a binary tree, determine if it is a valid binary search tree (BST).
Assume a BST is defined as follows:
The left subtree of a node contains only nodes with keys less than the node’s key. The right subtree of a node contains only nodes with keys greater than the node’s key. Both the left and right subtrees must also be binary search trees.

79. Valid Sudoku

Determine if a Sudoku is valid. A valid Sudoku board (partially filled) is not necessarily solvable. Only the filled cells need to be validated.

80. Verify Preorder Serialization of a Binary Tree

One way to serialize a binary tree is to use pre-order traversal. When we encounter a non-null node, we record the node’s value. If it is a null node, we record using a sentinel value such as #.

```
     _9_
    /   \
   3     2
  / \   / \
 4   1  #  6
/ \ / \   / \
# # # #   # #
```

For example, the above binary tree can be serialized to the string ″9,3,4,#,#,1,#,#,2,#,6,#,#″, where # represents a null node.
Given a string of comma separated values, verify whether it is a correct preorder traversal serialization of a binary tree. Find an algorithm without reconstructing the tree. Each comma separated value in the string must be either an integer or a character ’#’ representing null pointer. You may assume that the input format is always valid, for example it could never contain two consecutive commas such as ″1,,3″.
Example 1:
″9,3,4,#,#,1,#,#,2,#,6,#,#″
Return true
Example 2:
″1,#″
Return false
Example 3:
″9,#,#,1″
Return false

81. Word Search1

Given a 2D board and a word, find if the word exists in the grid.
The word can be constructed from letters of sequentially adjacent cell, where ″adjacent″ cells are those horizontally or vertically neighboring. The same letter cell may not be used more than once.

For example,

Given board =

```
[
  [’A’,’B’,’C’,’E’],
  [’S’,’F’,’C’,’S’],
  [’A’,’D’,’E’,’E’]
]
```

word = ″ABCCED″, -> returns true,

word = ″SEE″, -> returns true,

word = ″ABCB″, -> returns false.

82. Word Search2

Given a 2D board and a word, find if the word exists in the grid.

The word can be constructed from letters of sequentially adjacent cell, where ″adjacent″ cells are those horizontally or vertically neighboring. The same letter cell may not be used more than once.

For example,

Given board =

```
[
  [’A’,’B’,’C’,’E’],
  [’S’,’F’,’C’,’S’],
  [’A’,’D’,’E’,’E’]
]
```

word = ″ABCCED″, -> returns true,

word = ″SEE″, -> returns true,

word = ″ABCB″, -> returns false.

83. Word Subsets

We are given two arrays A and B of words. Each word is a string of lowercase letters.Now, say that word **b** is a subset of word **a** if every letter in b occurs in a, including multiplicity. For example, ″wrr″ is a subset of ″warrior″, but is not a subset of ″world″. Now say a word **a** from A is universal if for every **b** in B, **b** is a subset of **a**.

Return a list of all universal words in A. You can return the words in any order.

Example 1:

Input: A = [″amazon″,″apple″,″facebook″,″google″,″leetcode″], B = [″e″,″o″]

Output: [″facebook″,″google″,″leetcode″]

Example 2:

Input: A = [″amazon″,″apple″,″facebook″,″google″,″leetcode″], B = [″e″,″oo″]

Output: [″facebook″,″google″]

Example 3:

Input: A = [″amazon″,″apple″,″facebook″,″google″,″leetcode″], B = [″ec″,″oc″,″ceo″]

Output: [″facebook″,″leetcode″]

Note:

A[i] and B[i] consist only of lowercase letters.

All words in A[i] are unique: there isn’t i != j with A[i] == A[j].

84. ZigZag Order Level Traversal BST1

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

For example:

Given binary tree [3,9,20,null,null,15,7],

```
    3
   / \
  9  20
    /  \
   15   7
```

return its zigzag level order traversal as:

```
[
  [3],
  [20,9],
  [15,7]
]
```

85. ZigZag Order Level Traversal BST2

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

For example:

Given binary tree [3,9,20,null,null,15,7],

```
    3
   / \
  9  20
    /  \
   15   7
```

return its zigzag level order traversal as:

```
[
  [3],
  [20,9],
  [15,7]
]
```