**Algorithms Easy:**

**Find Numbers with Even Number of Digits**

Given an array nums of integers, return how many of them contain an even number of digits.

**Subtract the Product and Sum of Digits of an Integer**

Given an integer number n, return the difference between the product of its digits and the sum of its digits.

**Defanging an IP Address:**

Given a valid (IPv4) IP address, return a defanged version of that IP address.

A defanged IP address replaces every period "." with "[.]".

**Find N Unique Integers Sum up to Zero**

Given an integer n, return any array containing n unique integers such that they add up to 0.

**Maximum 69 Number**

Given a positive integer num consisting only of digits 6 and 9.

Return the maximum number you can get by changing **at most** one digit (6 becomes 9, and 9 becomes 6).

**Algorithms Medium:**

**Sum of Nodes with Even-Valued Grandparent**

Given a binary tree, return the sum of values of nodes with even-valued grandparent.  (A grandparent of a node is the parent of its parent, if it exists.)

If there are no nodes with an even-valued grandparent, return 0.

**Group the People Given the Group Size They Belong To**

There are n people whose **IDs** go from 0 to n - 1 and each person belongs **exactly** to one group. Given the array groupSizes of length n telling the group size each person belongs to, return the groups there are and the people's **IDs** each group includes.

You can return any solution in any order and the same applies for IDs. Also, it is guaranteed that there exists at least one solution.

**Deepest Leaves Sum**

Given a binary tree, return the sum of values of its deepest leaves.

**Insert into a Binary Search Tree**

Given the root node of a binary search tree (BST) and a value to be inserted into the tree, insert the value into the BST. Return the root node of the BST after the insertion. It is guaranteed that the new value does not exist in the original BST.

Note that there may exist multiple valid ways for the insertion, as long as the tree remains a BST after insertion. You can return any of them.

**All Elements in Two Binary Search Trees**

Given two binary search trees root1 and root2.

Return a list containing all the integers from both trees sorted in **ascending** order.

**Algorithms Hard:**

**Unique Paths III**

On a 2-dimensional grid, there are 4 types of squares:

* 1 represents the starting square.  There is exactly one starting square.
* 2 represents the ending square.  There is exactly one ending square.
* 0 represents empty squares we can walk over.
* -1 represents obstacles that we cannot walk over.

Return the number of 4-directional walks from the starting square to the ending square, that walk over every non-obstacle square exactly once.

**Maximum Score Words Formed by Letters**

Given a list of words, list of  single letters (might be repeating) and score of every character.

Return the maximum score of **any** valid set of words formed by using the given letters (words[i] cannot be used two or more times).

It is not necessary to use all characters in letters and each letter can only be used once. Score of letters 'a', 'b', 'c', ... ,'z' is given by score[0], score[1], ... , score[25] respectively.

**Minimum Number of Flips to Convert Binary Matrix to Zero Matrix**

Given a m x n binary matrix mat. In one step, you can choose one cell and flip it and all the four neighbours of it if they exist (Flip is changing 1 to 0 and 0 to 1). A pair of cells are called neighboors if they share one edge.

Return the minimum number of steps required to convert mat to a zero matrix or **-1** if you cannot.

Binary matrix is a matrix with all cells equal to 0 or 1 only.

Zero matrix is a matrix with all cells equal to 0.

**Number of Submatrices That Sum to Target**

Given a matrix, and a target, return the number of non-empty submatrices that sum to target.

A submatrix x1, y1, x2, y2 is the set of all cells matrix[x][y] with x1 <= x <= x2 and y1 <= y <= y2.

Two submatrices (x1, y1, x2, y2) and (x1', y1', x2', y2') are different if they have some coordinate that is different: for example, if x1 != x1'.