Permutations(Medium)
1. Given an array nums of distinct integers, return all the possible permutations. You can return the answer in any order.
Examples:
Input: nums = [1,2,3]
Output: [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]

Input: nums = [0,1]
Output: [[0,1],[1,0]]

Input: nums = [1]
Output: [[1]]

Constraints:
1 <= nums.length <= 6
-10 <= nums[i] <= 10
All the integers of nums are unique.
https://leetcode.com/problems/permutations/description/
~~~~~~
2. Next Permutation(Medium)
A permutation of an array of integers is an arrangement of its members into a sequence or linear order.

For example, for arr = [1,2,3], the following are all the permutations of arr: [1,2,3], [1,3,2], [2, 1, 3], [2, 3, 1], [3,1,2], [3,2,1].
The next permutation of an array of integers is the next lexicographically greater permutation of its integer. More formally, if all the permutations of the array are sorted in one container according to their lexicographical order, then the next permutation of that array is the permutation that follows it in the sorted container. If such arrangement is not possible, the array must be rearranged as the lowest possible order (i.e., sorted in ascending order).

For example, the next permutation of arr = [1,2,3] is [1,3,2].
Similarly, the next permutation of arr = [2,3,1] is [3,1,2].
While the next permutation of arr = [3,2,1] is [1,2,3] because [3,2,1] does not have a lexicographical larger rearrangement.
Given an array of integers nums, find the next permutation of nums. The replacement must be in place and use only constant extra memory.

Examples:
Input: nums = [1,2,3]
Output: [1,3,2]

Input: nums = [3,2,1]
Output: [1,2,3]

Input: nums = [1,1,5]
Output: [1,5,1]

Constraints:
1 <= nums.length <= 100
0 <= nums[i] <= 100
https://leetcode.com/problems/next-permutation/
~~~~~~
3. Permutation Sequence(Hard)
The set [1, 2, 3, ..., n] contains a total of n! unique permutations. By listing and labelling all of the permutations in order,
we get the following sequence for n = 3: "123", "132", "213", "231", "312", "321".
Given n and k, return the kth permutation sequence.

Examples:
Input: n = 3, k = 3
Output: "213"

Input: n = 4, k = 9
Output: "2314"

Input: n = 3, k = 1
Output: "123"

Constraints:
1 <= n <= 9
1 <= k <= n!
https://leetcode.com/problems/permutation-sequence/
~~~~~~
4. Permutations II(Medium)
Given a collection of numbers, nums, that might contain duplicates, return all possible unique permutations in any order.
~~~~~~

Examples:
Input: nums = [1,1,2]
Output: [[1,1,2], [1,2,1], [2,1,1]]

Input: nums = [1,2,3]
Output: [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]

Constraints:
1 <= nums.length <= 8
-10 <= nums[i] <= 10
https://leetcode.com/problems/permutations-ii/
~~~~~~
Combinations(Medium)
5. Given two integers n and k, return all possible combinations of k numbers chosen from the range [1, n].
You may return the answer in any order.

Examples:
Input: n = 4, k = 2
Output: [[1,2],[1,3],[1,4],[2,3],[2,4],[3,4]]
Explanation: There are 4 choose 2 = 6 total combinations.
Note that combinations are unordered, i.e., [1,2] and [2,1] are considered to be the same combination.

Input: n = 1, k = 1
Output: [[1]]
Explanation: There is 1 choose 1 = 1 total combination.

Constraints:
1 <= n <= 20
1 <= k <= n
https://leetcode.com/problems/combinations/
~~~~~~
Combination Sum(Medium)
6. Given an array of distinct integers candidates and a target integer target, return a list of all unique combinations of candidates where the chosen numbers sum to target. You may return the combinations in any order. The same number may be chosen from candidates an unlimited number of times. Two combinations are unique if the frequency of at least one of the chosen numbers is different. The test cases are generated such that the number of unique combinations that sum up to target is less than 150 combinations for the given input.

Examples:
Input: candidates = [2,3,6,7], target = 7
Output: [[2,2,3],[7]]
Explanation:
2 and 3 are candidates, and 2 + 2 + 3 = 7. Note that 2 can be used multiple times.
7 is a candidate, and 7 = 7. These are the only two combinations.

Input: candidates = [2,3,5], target = 8
Output: [[2,2,2,2],[2,3,3],[3,5]]

Input: candidates = [2], target = 1
Output: []

Constraints:
1 <= candidates.length <= 30
2 <= candidates[i] <= 40
All elements of candidates are distinct.
1 <= target <= 40
https://leetcode.com/problems/combination-sum/
~~~~~~
7. Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent. Return the answer in any order. A mapping of digits to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.

 1   2   3
    abc def

 4   5   6
ghi jkl mno

 7   8   9
pqrs tuv wxyz
~~~~~~

Examples:
Input: digits = "23"
Output: ["ad","ae","af","bd","be","bf","cd","ce","cf"]

Input: digits = ""
Output: []

Input: digits = "2"
Output: ["a","b","c"]

Constraints:
0 <= digits.length <= 4
digits[i] is a digit in the range ['2', '9']
https://leetcode.com/problems/letter-combinations-of-a-phone-number/
~~~~~~

8. Sort Colors(Medium)
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.

We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively.

You must solve this problem without using the library's sort function.

Example 1:
Input: nums = [2,0,2,1,1,0]
Output: [0,0,1,1,2,2]

Example 2:
Input: nums = [2,0,1]
Output: [0,1,2]

Constraints:
n == nums.length
1 <= n <= 300
nums[i] is either 0, 1, or 2.

Follow up: Could you come up with a one-pass algorithm using only constant extra space?
https://leetcode.com/problems/sort-colors/description/
~~~~~~

9. There are ‘p’ balls of type P, ‘q’ balls of type Q and ‘r’ balls of type R. Using the balls we want to create a straight line such that no two balls of same type are adjacent.

Examples :
Input : p = 1, q = 1, r = 0
Output : 2
There are only two arrangements PQ and QP

Input : p = 1, q = 1, r = 1
Output : 6
There are only six arrangements PQR, QPR,
QRP, RQP, PRQ and RPQ

Input : p = 2, q = 1, r = 1
Output : 12
There are twelve arrangements PQRP, PRQP, QPRP, RPQP, PQPR, PRPQ
QPQR, RQRP, PQRQ, QRQP, RPQR, and QRPR
~~~~~~

9. Minimum Number of Operations to Move All Balls to Each Box(Medium)
You have n boxes. You are given a binary string boxes of length n, where boxes[i] is '0' if the ith box is empty, and '1' if it contains one ball.
In one operation, you can move one ball from a box to an adjacent box. Box i is adjacent to box j if abs(i - j) == 1. Note that after doing so, there may be more than one ball in some boxes.
Return an array answer of size n, where answer[i] is the minimum number of operations needed to move all the balls to the ith box.
Each answer[i] is calculated considering the initial state of the boxes.

Examples:
Input: boxes = "110"
Output: [1,1,3]
Explanation: The answer for each box is as follows:
1) First box: you will have to move one ball from the second box to the first box in one operation.
~~~~~~

2) Second box: you will have to move one ball from the first box to the second box in one operation.
3) Third box: you will have to move one ball from the first box to the third box in two operations, and move one ball from the second box to the third box in one operation.

Input: boxes = "001011"
Output: [11,8,5,4,3,4]

Constraints:
n == boxes.length
1 <= n <= 2000
boxes[i] is either '0' or '1'.
https://leetcode.com/problems/minimum-number-of-operations-to-move-all-balls-to-each-box/description/
~~~~~~

10. Given an array A consisting of N positive integers, find the total number of subsequences of the given array such that the chosen subsequence represents a permutation.

Note: Sequence A is a subsequence of B if A can be obtained from B by deleting some(possibly, zero) elements without changing its order. For example, [3,1] is a subsequence of [3,2,1] and [4,3,1], but not a subsequence of [1,3,3,7] and [3,10,4].
Two permutations are different if they are of different lengths or if there is any element in the two permutations such that its index in the original array is different.
A permutation of length N is an array of length N in which every element from 1 to N occurs exactly once.
Examples:
Input: N = 5, A[] = {1, 2, 3, 2, 4}
Output: 7
Explanation: We can get 7 permutations: {1}, {1, 2}, {1, 2}, {1, 2, 3}, {1, 2, 3}, {1, 2, 3, 4}, {1, 2, 3, 4}.

Input: N = 4, A[] = {2, 1, 1, 2}
Output: 6
Explanation: We can get 6 permutations: {1}, {1}, {2, 1}, {2, 1}, {1, 2}, {1, 2}
https://www.geeksforgeeks.org/count-permutations-in-a-sequence/?ref=ml_lbp
~~~~~~

11. Maximum Number of Balls in a Box(Easy)
You are working in a ball factory where you have n balls numbered from lowLimit up to highLimit inclusive (i.e., n == highLimit - lowLimit + 1), and an infinite number of boxes numbered from 1 to infinity. Your job at this factory is to put each ball in the box with a number equal to the sum of digits of the ball's number. For example, the ball number 321 will be put in the box number 3 + 2 + 1 = 6 and the ball number 10 will be put in the box number 1 + 0 = 1.
Given two integers lowLimit and highLimit, return the number of balls in the box with the most balls.

Examples:
Input: lowLimit = 1, highLimit = 10
Output: 2
Explanation:
Box Number: 1 2 3 4 5 6 7 8 9 10 11 ...
Ball Count: 2 1 1 1 1 1 1 1 1 0 0 ...
Box 1 has the most number of balls with 2 balls.

Input: lowLimit = 5, highLimit = 15
Output: 2
Explanation:
Box Number: 1 2 3 4 5 6 7 8 9 10 11 ...
Ball Count: 1 1 1 1 2 2 1 1 1 0 0 ...
Boxes 5 and 6 have the most number of balls with 2 balls in each.

Input: lowLimit = 19, highLimit = 28
Output: 2
Explanation:
Box Number: 1 2 3 4 5 6 7 8 9 10 11 12 ...
Ball Count: 0 1 1 1 1 1 1 1 1 2 0 0 ...
Box 10 has the most number of balls with 2 balls.

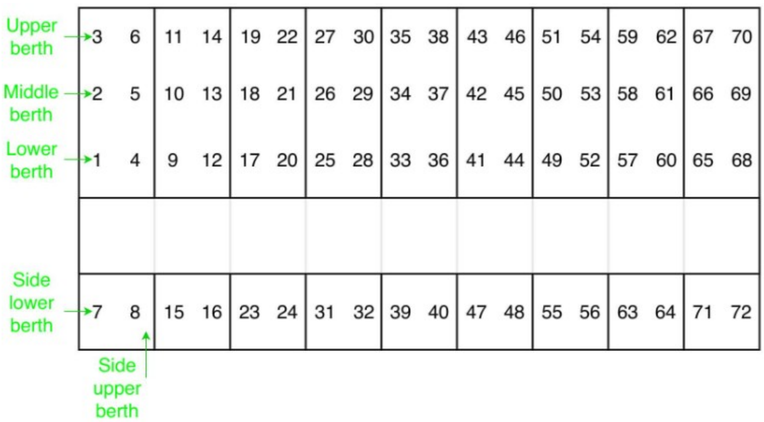
Constraints:
1 <= lowLimit <= highLimit <= 10^5
https://leetcode.com/problems/maximum-number-of-balls-in-a-box/description/
~~~~~~

12. Find the Number of Distinct Colors Among the Balls(Medium)
You are given an integer limit and a 2D array queries of size n x 2. There are limit + 1 balls with distinct labels in the range [0, limit]. Initially, all balls are uncolored. For every query in queries that is of the form [x, y], you mark ball x with the color y. After each query, you need to find the number of distinct colors among the balls. Return an array result of length n, where result[i] denotes the number of distinct colors after ith query.
Note that when answering a query, lack of a color will not be considered as a color.
~~~~~~

Example 1:
Input: limit = 4, queries = [[1,4],[2,5],[1,3],[3,4]]
Output: [1,2,2,3]
Explanation:
After query 0, ball 1 has color 4.
After query 1, ball 1 has color 4, and ball 2 has color 5.
After query 2, ball 1 has color 3, and ball 2 has color 5.
After query 3, ball 1 has color 3, ball 2 has color 5, and ball 3 has color 4.

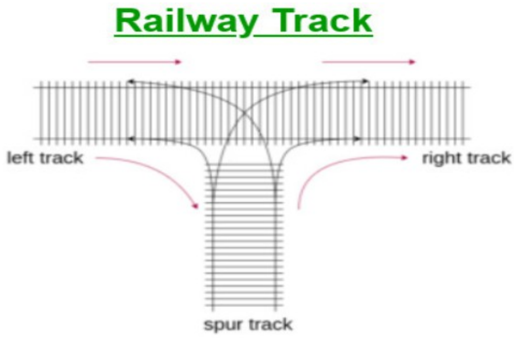
Example 2:
Input: limit = 4, queries = [[0,1],[1,2],[2,2],[3,4],[4,5]]
Output: [1,2,2,3,4]
Explanation:
After query 0, ball 0 has color 1.
After query 1, ball 0 has color 1, and ball 1 has color 2.
After query 2, ball 0 has color 1, and balls 1 and 2 have color 2.
After query 3, ball 0 has color 1, balls 1 and 2 have color 2, and ball 3 has color 4.
After query 4, ball 0 has color 1, balls 1 and 2 have color 2, ball 3 has color 4, and ball 4 has color 5.

Constraints:
1 <= limit <= 10^9
1 <= n == queries.length <= 10^5
queries[i].length == 2
0 <= queries[i][0] <= limit
1 <= queries[i][1] <= 10^9
https://leetcode.com/problems/find-the-number-of-distinct-colors-among-the-balls/
\~~~~~~
13. Given a railway seat number, the task is to check whether it is a valid seat number or not. Also print its berth type i.e lower berth, middle berth, upper berth, side lower berth, side upper berth as per the figure below.



Examples:
Input: 10
Output: middle berth

Input: 7
Output: side lower berth
https://www.geeksforgeeks.org/program-to-print-the-berth-of-given-railway-seat-number/
\~~~~~~
Possible permutations at a railway track
14. Given a left, right, and a spur track as shown in the below figure. There are N trucks from value 1 to N arranged in the left track. We can move directly N trucks to the right track but there can be more possibilities of moving the trucks to the right track using the spur track. We can move any truck to spur track and then move it to the right track. The task is to print all the possible permutation order in which all the N trucks can be moved from left track to right track. Note: Once a truck is moved from left truck to right/spur track then it can’t be moved to left track again.



Examples:
Input: N = 2
Output: 1 2 2 1
Explanation:
For the first permutation: left[] = {1, 2} right[] = {}, and spur[] = {} The truck with value 2 moved to the right track, then left[] = {1} right[] = {2}, and spur[] = {}

Now moving with value 1 to the right track, then left[] = {} right[] = {1, 2}, and spur[] = {}
For the second permutation: left[] = {1, 2} right[] = {}, and spur[] = {} The truck with value 2 move to the spur track, then left[] = {1} right[] = {}, and spur[] = {2}
The truck with value 1 move to the right track, then left[] = {} right[] = {1}, and spur[] = {2}
The truck with value 2 in the spur track move to the right track, then left[] = {} right[] = {2, 1}, and spur[] = {}

Input: N = 3
Output: 1 2 3 2 1 3 3 2 1 3 1 2 2 3 1
https://www.geeksforgeeks.org/possible-permutations-at-a-railway-track/
~~~~~~

15. Given the arrival and departure times of all trains that reach a railway station, the task is to find the minimum number of platforms required for the railway station so that no train waits. We are given two arrays that represent the arrival and departure times of trains that stop.

Examples:
Input: arr[] = {9:00, 9:40, 9:50, 11:00, 15:00, 18:00},
dep[] = {9:10, 12:00, 11:20, 11:30, 19:00, 20:00}
Output: 3
Explanation: There are at-most three trains at a time (time between 9:40 to 12:00)

Input: arr[] = {9:00, 9:40}, dep[] = {9:10, 12:00}
Output: 1
Explanation: Only one platform is needed.

https://www.geeksforgeeks.org/minimum-number-platforms-required-railwaybus-station/
~~~~~~