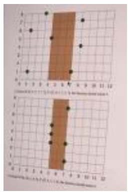
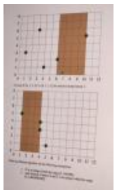
CODILITY

03 Tháng Chín 2024 10:24 SA

!!! O(NlogN) trở xuống !!!

TASK 1

1. TheWidestPath
Given a forest containing N trees, find the width of the widest vertical path that can be built without cutting any tree.
Task description:
- There are N trees (numbered from 0 to N-1) in a forest. The K-th tree is located at coordinates (X[K], Y[K]).
- We want to build the widest possible vertical path, such that there is no tree on it. The path must be built somewhere between a leftmost and a rightmost tree, which means that the width of the path cannot be infinite.
- What is the width of the widest possible path that can be built?
- Write a function: function solution(X, Y); that, given two arrays X and Y consisting of N integers each, denoting the positions of trees, returns the width of the widest possible path that can be built.
- Examples:
  - 1. Given X=[1, 8, 7, 3, 4, 1, 8], Y=[6, 4, 1, 8, 5, 1, 7], the function should return 3.

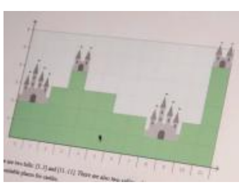



  - There are no 2 two with the same coordinates
  - A path of width at least 1 can always be built

2. CardPayments
Given a list of transactions within one year, calculate the final balance of an account. Add a fee for each month that did not include at least three card payments for a total sum of at least 100.
Task description:
- You are given a list of all the transactions on a bank account during the year 2020. The account was empty at the beginning of the year (the balance was 0).
- Each transaction specifies the amount and the date it was executed. If the amount is negative (less than 0) then it was a card payment, otherwise it was an incoming transfer (amount at least 0). The date of each transaction is in YYYY-MM-DD format: for example, 2020-05-20 represents 20th May 2020.
- Additionally, there is a fee for having a card (omitted in the given transaction list), which is 5 per month. This fee is deducted from the account balance at the end of each month unless there were at least three payments made by card for a total cost of at least 100 within that month.
- Your task is to compute the final balance of the account at the end of the year 2020.
- Write a function: function solution(A, D); that, given an array A of N integers representing transaction amounts and an array D of N strings representing transaction dates, returns the final balance of the account at the end of the year 2020. Transaction number K (for K within the range [0..N-1]) was executed on the date represented by D[K] for amount A[K].
- Examples:
  - Given A = [100, 100, 100,-10] and D= ["2020-12-31", "2020-12-22", "2020-12-03", "2020-12-29"], the function should return 230. Total income was equal to 100 + 100 -+ 100 - 10 =290 and the fee was paid every month, so 290- (5 * 12) = 230.
  - Given A = [180, -50, -25, -25] and D = ["2020-01-01", "2020-01-01", "2020-01-01". "2020-01-31"], the function should return 25. The income was equal to 180, the expenditure was equal to 100 and the fee was applied in every month except January: 180- 100- (5 * 11) = 25
  - Given A = [1,-1, 0, -105, 1] and D = ["2020-12-31", "2020-04-04", "2020-04-04", "2020-04-14", "2020-07-12"), the function should return -164. The fee is paid every month: 1 - 1 + 0 - 105 + 1 - (5 * 12) = -164. Note that in April, even though the total cost of card payments was 106 (more than 100), there were only two payments made by card, so the fec was still applied. A transaction of value 0 is considered a positive, incoming transfer.
  - Given A = [100, 100, -10, -20, -30] and D = ["2020-01-01", "2020-02-01", "2020-02-11". "2020-02-05", "2020-02-08"], the function should return 80.
  - Given A = [-60, 60, -40, -20] and D = ["2020-10-01", "2020-02-02", "2020-10-10" "2020-10-30"], the function should return -115.
  - Assume that:
    - N is an integer within the range [1..100];
    - each element of array A is an integer within the range [-1,000..1,000];
    - D contains strings in YYYY-MM-DD format, representing dates in the range 2020-01-01 to 2020-12-31.

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

3. CastleBuilding
Find the number of castles that can be built.
Task description:
- Charlemagne, the King of Frankia, is considering building some castles on the border with Servia. The border is divided into N segments.
- The King knows the height of the terrain in each segment of the border. The height of each segment of terrain is stored in array A, with A[P] denoting the height of the P-th segment of the border. The King has decided to build a castle on top of every hill and in the bottom of every valley.
- Let [P..Q] denote a group of consecutive segments from P to Q inclusive such that (0<=P<=Q<=N-1). Segments [P..Q] form a hill or a valley if all the following conditions are satisfied:
  - The terrain height of each segment from P to Q is the same (A[P] = A[P+1] = .. = A[Q];
  - If P > 0 then A[P-1] < A[P] (for a hill) or A[P- 1] > A[P] (for a valley);
  - If Q < N-1 then A[Q+1] < A[Q] (for a hill) or A[Q+1]> A[Q] (for a valley);
- That is, a hill is higher than its surroundings and a valley is lower than its surroundings. Note that if the suroundings on either side of the hill or valley don't exist (i.e. at the edges of the area under consideration, where P = 0 or Q = N-1), then the condition is considered satisfied for that side of the hill/valley.
- The king is wondering how many castles is he going to build. Can you help him?
- For example, consider the following array A = [2, 2, 3, 4, 3, 3, 2, 2, 1, 1, 2, 5].



- There are two hills: [3..3] and [11..11]. There are also two valleys: [0..1] and [8..9]. There are no other suitable places for castles.
- Write a function: function solution(A); that, given an array A consisting of N integers, as explained above, returns the total number of hills and valleys.
- For example, given array A as described above, the function should return 4.
- Given array A = [-3, -3] describing segments with a terrain height below 0, segment [0..1] forms both a hill and a valley, and only one castle can be built, so the function should return 1.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - each element of array A is an integer within the range [-1,000,000,000..1,000,000,000].

4. CommonLetter
Given an array of strings, find a pair of strings that share the same letter at the same position.
Task description:
- You are given an array S consisting of N strings. Every string is of the same length M. Your task is to find a pair of strings in array S, such that there exists a position in which both of the strings have the same letter. Both the index in array S and the positions in the strings are numbered from zero.
- For example, given S = ["abc", "bca","dbe"], string 0 ("abc") and string 2 ("dbe") have the same letter 'b' in position 1. On the other hand for strings "abc" and "bca" there does not exist a position in which they have the same letter.
- Write a function: function solution(S); that, given a zero-indexed array S of N strings, returns an array describing a pair of strings from S which share a common letter at some index. If there is no such pair, the function should return an empty array. If there is more than one correct answer, the function can return any of them.
- The result should be represented as an array containing three integers. The first two integers are the indexes in S of the strings belonging to the pair. The third integer is the position of the common letter.
- For S ["abc", "bca", "dbe"], as above, the result array should be represented as [0, 2, 1]. Another correct answer is [2, 0, 1], as the order of indexes of strings not matter.
- Examples:
  - Given: S = ["abc", "bca", "dbe"], your function may return [0, 2, 1] as described above,
  - Given: S = ["zzzz", "ferz", "zdsr", "fgtd"], your function may return [0, 1, 3]. Both "zzzz" and "ferz" have 'z' in position 3. The function may also return [1, 3, 0], which would reflect strings "ferz" "fgtd" and letter 'f'

TASK 2

1. HolidayTrip
Calculate the minimum number of cars needed to take all of the passengers.
Task description:
- A group of friends is going on holiday together. They have come to a meeting point (the start of the journey) using N cars. There are P[K] people and S[K] seats in the K-th car for K in range [0..N-1]. Some of the seats in the cars may be free, so it is possible for some of the friends to change the car they are in. The friends have decided that, in order to be ecological, they will leave some cars parked at the meeting point and travel with as few cars as possible.
- Write a function: function solution(P, S); that, given two arrays P and S, consisting of N integers each, returns the minimum number of cars needed to take all of the friends on holiday.
- Examples:
  - Given P=[1,4, 1] and S = [1, 5, 1], the function should return 2. A person from car number 0 can travel in car number 1 instead. This way, car number 0 can be left parked at the meeting point.
  - Given P= [4, 4, 2, 4] and S = [5, 5, 2, 5], the function should return 3. One person from car number 2 can travel in car number 0 and the other person from car number 2 can travel in car number 3.
  - Given P= [2,3, 4, 2] and S = [2, 5, 7, 2], the fiunction should return 2. Passengers from car number0 can travel in car number 1 and passengers from car number 3 can travel in car number 2.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - each element of arrays P and S is an integer within the range [1..9];
  - every friend had a seat in the car they came in; that is, P[K] <= S[K] for each K within the range [0..N-1]

2. Missions
Calculate the minimum number of days required to complete all of the missions in a game.
Task description:
- In order to finish a game, a player has to complete N missions. The missions are numbered from 0 to N-1. The K-th mission has an integer D[K] assigned, representing its dificulty level.
- During a day, you can perform any number of missions given the two following rules:
  - Missions should be performed in the specified order, in other words, a mission can be undertaken only if all of the misions preceding it have already been completed;
  - The diffrence between the difficulty levels of any two missions performed on the same day should not be greater than an integer X.
- Write a function: function solution(D, X); that, given an array D of N integers and an integer X, returns the minimum number of days required to complete all of the missions in the game.
- Examples:
  - Given D = [5, 8, 2, 7] and X = 3, your function should return 3. The first two missions can be performed on the first day, the third mission on the second day and the last mission on the third day. It is not possible to complete all of the missions in fewer days.
  - Given D = [2, 5, 9, 2, 1, 4] and X = 4, your function should return 3. The first two missions can be performed on the first day, the third mission on the second day and all of the remaining missions on the third day. Note that it is possible to perform the first mission on the first day and the next two missions on the second day. In both of these cases, the minimum number of days required to complete all of the missions is 3.
  - Given D= [1, 12, 10, 4, 5, 2] and X = 2, your fiunction should return 4. The first mission can be performed on the first day, the next two missions on the second day, the fourth and fifth missions on the third day, and the last remaining mission on the fourth day. It is not possible to complete all of the missions in fewer days.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..200,000];
  - X is an integer within the range [0..1,000,000,000];
  - each element of array D is an integer within the range [1..1,000,000,000].

3. PlayersMovements
Calculate the number of players that willperform a move.
Task description:
- There are N players standing in a row, one player on a field. They are numbered from 0 to N-1 from left to right.
- Players perfom moves one by one from left to right, that is, in ascending order of numbers. Each player presses an arrow key in one of the four cardinal directions: left ('<'), right ('>'), up ('^') or down ('v'). A key press in the given direction means that the player attempts to move onto the closest field in the direction specified. A move can be performed only if there is no other player already standing on the target field.
- Moves are represented as a string S of length N, where S[K] (for K within the range 0..N-1) is the direction of the K-th player's move. How many players will actually perform a move successfully?
- Write a function: function solution(S); which, given a string S of length N representing arrow keys pressed by each of the players, returns the number of players that will perform a move successfully.
- Examples:
  - Given S= "><^v", your function should return 2. Player 0 cannot move rightwards, because player 1 is standing on the target field. Player 1 cannot move leftwards, because player 0 is standing on the target field. Players 2 and 3 can both perform their moves because there are no other players standing on their target fields. In the pictures below, players that will perform their moves successflly are marked green and players whose moves will fail are marked red.
  - Given S = "<<^<v>>", your function should retun 6. Players 0 1, 2, 3, 4 can all perform their moves because there are no other players standing on their target fields. Player 5 pressed the right-arrow key, but the move cannot be performed as player 6 is already standing on the target field. The move of player 6 can be performed, though, as there is no other player standing on the target field.
  - Given S = "><><", your function should return 0. No player can perform a move.
- Assume that:
  - N is an integer within the range [1..50];
  - string S is made only of the following characters: '^', 'v', '<' and/or '>`.

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

4. SortedTwoLettersWord
Given a string, count the minimum number of letters that need to be deleted in order to obtain a string in the format "A...AB...B".
Task description:
- We are given a string S of length N consisting only of letters 'A' and/or 'B'. Our goal is to obtain a string in the format "A...AB...B" (all letters 'A' occur before all letters 'B) by deleting some letters from S. In particular, strings consisting only of letters 'A' or only of letters 'B' fit this format.
- Write a function: function solution(S); that, given a string S, returns the minimum number of letters that need to be deleted from S in order to obtain a string in the above format.
- Examples:
  - Given S = "BAAABAB", the function should return 2. We can obtain "AAABB" by deleting the first occurence of 'B' and the last occurrence of 'A".
  - Given S = "BBABAA", the function should return 3. We can delete all occurrences of 'A' or all occurrences of 'B'.
  - Given S = "AABBBB", the function should return 0. We do not have to delete any letters, because the given string is already in the expected format.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range[1..100,000];
  - string S is made only of the characters 'A ' and/or 'B'.

TASK 3

1. DiceRolls
Given a subset of a series of dice roll results, and the arithmetic mean of all results, find the missing results.
Task description:
- You have just rolled a dice several times. The N roll results that you remember are described by an array A. However, there are F rolls whose results you have forgotten. The arithmetic mean of all of the roll results (the sum of all the roll results divided by the number of rolls) equals M.
- What are the possible results of the mising rolls?
- Write a function: function solution(A, F, M); that, given an array A of length N, an integer F and an integer M, returns an array containing possible results of the missed rolls. The returned array should contain F integers from 1 to 6 (valid dice rolls). If such an array does not exist then the function should return [0].
- Examples:
  - Given A = [3, 2, 4, 3], F = 2, M = 4, your function should return [6, 6]. The arithmetic mean of all the rolls is (3+2+4+3+6+6) / 6 = 24 / 6 = 4
  - Given A = [1, 5, 6], F = 4, M = 3, your function may return [2, 1, 2, 4] or [6, 1, 1, 1] (among others).
  - Given A =[1, 2, 3, 4], F = 4, M = 6, your function should return [0]. It is not possible to obtain such a mean.
  - Given A= [6, 1], F = 1, M = 1, your function should return [0]. It is not possible to obtain such a mean.
- Write an efficient algorithm for the following assumptions:
  - N and F are integer within the range [1..100,000];
  - each element of arrays A is an integer within the range [1..6];
  - M is an integer within the range [1..6]

Remember, all submissions are being check for plagiarism. Your recruiter will be informed in case suspicious activity is detected.

2. DistinctNumbersCount
Count the minimum number of integers that must be deleted from an array so that no two integers occur the same number of times.
Task description:
- An array A consisting of N integers is given. Our goal is to obtain an array in which every value occurs a unique number of times. We only consider integers that appear at least once in the resulting array. To achieve the goal, we can delete some integers from A. What is the minimum number of integers that must be delcted from A so that every remaining value occurs unique number of times?
- Write a function: function solution(A); that, given an array A consisting of N integers, returns the minimum number of integers that must be deleted from it so that every remaining value occurs a unique number of times.
- Example:
  - Given A = [1, 1, 1, 2, 2, 2], the function should return 1. We can delete one occurrence of 1 or one occurrence of 2. After this operation, one value will occur three times and the other one two times.
  - Given A = [5, 3, 3, 2, 5, 2, 3, 2], the function should return 2. After deleting number 3 twice, the remaining elements of the array are [5, 2, 5, 2, 3, 2]. In this array no two numbers occur the same number of times.
  - Given A = [127, 15, 3, 8, 10], the function should return 4. All elements of the array occur exactly once. We have to delete all but one element.
  - Give A = [10000000, 10000000, 5, 5, 5, 2, 2, 2, 0, 0], the function should return 4.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..200,000];
  - each element of array A is an integer within the range [1..1,000,000,000].

3. BankTransfer
Given a list of transfers between two banks, determine the minimum initial account balance of each bank required for the given transfers to be carried out.
Task description:
- You are given a list of N transfers (numbered from 0 to N-1) between two banks: bank A and bank B. The K-th transfer is described by two values:
  - R[K] (either "A" or "B") representing the recipient (the bank the transfer is sent to);
  - V[K] denoting the value sent via the transfer.
- All transfers are completed in the order they appear on the list. The banks do not want to go into debt (in other words, their account balance may not drop below 0). What minimum initial account balance in each bank is necessary in order to complete the transfers?
- Write a function: function solution(R, V); that, given a string R and an array of integers V, both of length N, returns an array of two integers. The integers should represent the minimum initial account balances for banks A and B in the following order: [bank A, bank B].
- Result array should be retured as an array of integers.
- Examples:
  - Given R = "BAABA" and V = [2, 4, 1, 1, 2], the function should return [2, 4]. The bank accounts balances after each transfer are shown in the following table:

| A | B ------------------------+---+--- initial balance | 2 | 4 transfer 2 from A to B | 0 | 6 transfer 4 from B to A | 4 | 2 transfer 1 from B to A | 5 | 1 transfer 1 from A to B | 4 | 2 transfer 2 from B to A | 6 | 0

  - Given R = "ABAB" and V = [10, 5, 10, 15], the function should return [0, 15].
  - Given R = "B" and V = [100], the function should return [100, 0].
- Write an efficient algorithm for the following assumptions:
  - string R and array V are both of length N;
  - N is an integer within the range [1..100,000];
  - each element of array V is an integer within the range [1..10,000];
  - string R is made only of the character 'A' and/or 'B'.

4. MaxSwitchingSlice
Given an array, find the length of the longest switching slice.
Task description:
- We call an array switching if all numbers in even positions are equal and all numbers in odd positions are equal.
- For example: [3, -7, 3, -7, 3] and [4, 4, 4] are switching, but [5, 5, 4, 5, 4] and [-3, 2, 3] are not switching.
- What is the length of the longest switching slice (continuous fragment) in a given array A?
- Write a function: function solution(A); that, given an array A consisting of N integers, returns the length of the longest switching slice in A.
- Examples:
  - Given A = [3, 2, 3, 2, 3], the function should return 5, because the whole array is switching.
  - Given A = [7, 4, -2, 4,-2, -9], the function should return 4. The longest switching slice is [4, -2, 4, -2].
  - Given A =[7,-5,-5, -5, 7, -1, 7], the function should return 3. There are two switching slices of equal length: [-5, -5, -5] and [7, -1, 7].
  - Given A = [4], the function should return 1. A single-element slice is also a switching slice.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range[1..100,000];
  - each element of array A is an integer within the range [-1,000,000,000...1,000,000,000].

5. BattleshipRecognition
Given a rectangular board containing ships of three different sizes, find the number of ships of each size.
Task description:
- Battleships is a game played on a rectangular board. You are given a representation of such a board of size N (height) x M (width) with information about the locations of the ships.
- The board is given as an array B, whose every element is a string that corresponds to one row of S the game board. Each character each string represents cell oft the board and is either:
  - a '#' character, marking a part of a ship; or
  - a '.' character, representing an empty cell.
- Two cells that share a side and have a value of '#' are parts of the same ship. Cell (X, Y) shares a side with cells (X, Y-1), (X, Y+I), (X-1, Y) and (X+1, Y).
- In the Batleships game there are three types of ships:
  - Patrol Boats of size 1 (a square)
  - Submarines of size 2, which come in two shapes
  - Destroyers of size 3, which come in six shapes
- Your task to find the number of ships of each type occurring on the board.
- For example, on the board represented by B = [". # # . # ", "# . # . .", "# . . . #", "# . # # ."], there are two patrol boats (marked on the image below as 'P'), one submarine ('S') and two destroyers ('D').
- Write a function: function solution(B); that, given an array B consisting of N strings of length M each, returns an array R of three integers, such that:
  - R[0] represents the number of Patrol Boats,
  - R[1] represents the number of Submarines,
  - R[2] represents the number of Destroyers.
- Examples:
  - Given B = [". # # . #". "# . # . .", "# . . . #", "# . # # ."] the function should return [2, 1, 2] as explained above.
  - Given B=[". # . . #", "# # . . #", ". . . # ."], the function should return [1, 1, 1].
  - Given B= ["# # .", "# . #", ". # #"], the fiunction should return [0, 0, 2].
  - Given B = [". . .", ". . .", ". . ."], the function should return [0, 0, 0].
- Assume that:
  - N is an integer within the range [1..100];
  - all strings in B are of the same length M from the range [1.100];
  - every string in B consists only of the following characters: '.' and/or '#';
  - every ship on the board is either a Patrol Boat (size 1), a Submarine (size 2) or a Destroyer (size 3).

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

6. SanatoriumAccommodation
Find the minimum number of rooms needed to accommodate all guests in a sanatorium according to their preferences.
Task description:
- There are N guests (numbered from 0 to N-1) in a sanatorium waiting to be assigned a room. In each room, any number of guests can be

```
public class LongestCommonPrefix {
    public String longestCommonPrefix(String[] strs) {
        // If the array is empty, return an empty string
        if (strs == null || strs.length == 0) {
            return "";
        }

        // Start with the first string as the prefix
        String prefix = strs[0];

        // Compare the prefix with each string in the array
        for (int i = 1; i < strs.length; i++) {
            // Update the prefix by comparing it with the current string
            while (strs[i].indexOf(prefix) != 0) {
                // Reduce the prefix length by one character
                prefix = prefix.substring(0, prefix.length() - 1);
                // If the prefix becomes empty, return an empty string
                if (prefix.isEmpty()) {
                    return "";
                }
            }
        }

        return prefix;
    }

    public static void main(String[] args) {
        LongestCommonPrefix lcp = new LongestCommonPrefix();
        String[] strs1 = {"flower", "flow", "flight"};
        String[] strs2 = {"dog", "racecar", "car"};

        System.out.println("Longest Common Prefix of strs1: " + lcp.longestCommonPrefix(strs1)); // Output: "fl"
        System.out.println("Longest Common Prefix of strs2: " + lcp.longestCommonPrefix(strs2)); // Output: ""
    }
}
```

```
public class LongestCommonPrefix {
    public static String longestCommonPrefix(String[] strs) {
        if (strs == null || strs.length == 0) {
            return "";
        }

        // Duyệt qua từng ký tự của chuỗi đầu tiên
        for (int i = 0; i < strs[0].length(); i++) {
            char c = strs[0].charAt(i);

            // So sánh ký tự này với các chuỗi còn lại
            for (int j = 1; j < strs.length; j++) {
                // Nếu đến cuối chuỗi hoặc ký tự khác nhau, trả về phần tiền tố đã tìm được
                if (i >= strs[j].length() || strs[j].charAt(i) != c) {
                    return strs[0].substring(0, i);
                }
            }
        }

        // Nếu chuỗi đầu tiên là tiền tố của tất cả các chuỗi, trả về chuỗi đầu tiên
        return strs[0];
    }

    public static void main(String[] args) {
        // Test case 1
        String[] strs1 = {"flower", "flow", "flight"};
        System.out.println("Longest Common Prefix: " + longestCommonPrefix(strs1)); // Output:
"fl"

        // Test case 2
        String[] strs2 = {"dog", "racecar", "car"};
        System.out.println("Longest Common Prefix: " + longestCommonPrefix(strs2)); // Output:
""
    }
}
```

```
public class LongestCommonPrefix {
    public static String longestCommonPrefix(String[] strs) {
        if (strs == null || strs.length == 0) {
            return "";
        }

        // Start by assuming the whole first string is the common prefix
        String prefix = strs[0];

        // Compare the prefix with each string in the array
        for (int i = 1; i < strs.length; i++) {
            // Adjust the prefix until it matches the current string
            while (strs[i].indexOf(prefix) != 0) {
                // Remove the last character from the prefix
                prefix = prefix.substring(0, prefix.length() - 1);
                // If the prefix becomes empty, return an empty string
                if (prefix.isEmpty()) {
                    return "";
                }
            }
        }

        return prefix;
    }

    public static void main(String[] args) {
        // Test cases
        String[] strs1 = {"flower", "flow", "flight"};
        System.out.println("Longest Common Prefix: " + longestCommonPrefix(strs1)); // Output: "fl"

        String[] strs2 = {"dog", "racecar", "car"};
        System.out.println("Longest Common Prefix: " + longestCommonPrefix(strs2)); // Output: ""
    }
}
```

position of the common letter.
- For S ["abc", "bca", "dbe"], as above, the result array should be represented as [0, 2, 1]. Another correct answer is [2, 0, 1], as the order of indexes of strings not matter.
- Examples:
  - Given: S = ["abc", "bca", "dbe"], your function may return [0, 2, 1] as described above,
  - Given: S = ["zzzz", "ferz", "zdsr", "fgtd"], your function may return [0, 1, 3]. Both "zzzz" and "ferz" have 'z' in position 3. The function may also return [1, 3, 0], which would reflect strings "ferz","fgtd" and letter 'f'.
  - Given A = ["gr","sd", "rg", your function should return []. There is no pair of strings that fulfils the criteria.
  - Given A = ["bdafg", "ceagi"], your function may return [0, 1, 2].
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..30,000];
  - M is an integer within the range [1..2,000];
  - each element of S consists only of lowercase English letters (a-z);
  - N * M <= 30,000.

5. CountBananas

Calculate how many times you can print the word 'BANANA" using the letters given in string S.

Task description:

- A string S made of uppercase English letters is given. In one move, six letters forming the word "BANANA" (one 'B', three 'A's and two 'N's) can be deleted from S. What is the maximum number times such a move can be applied to S?
- Write a function: function solution(S); that, given a string S of length N, returns the maximum number of moves that can be applied.
- Examples:
  - Given S = "NAABXXAN", the function should return 1.
  - Given S = "NAANAAXNABABYNNBZ", the function should return 2.
  - Given S = "QABAAAAWOBL", the function should return 0.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - string S is made only of uppercase letters (a-z).

6. CreatePalindrome

Replace all question marks in a given string to obtain a palindrome.

Task description:

- Write a function solution that, given a string S of length N, returns any palindrome which can be obtained by replacing all of the question marks in S by lowercase letters ('a' - 'z'). If no palindrome can be obtained, the function should return the string "NO".
- A palindrome is a string that reads the same both forwards and backwards. Some examples of palindromes are: "kayak", "radar", "mom".
- Examples:
  - Given S = "?ab??a", the function should retum "aabbaa"
  - Given S = "bab??a", the function should return "NO".
  - Given S = "?a?", the function may return "aaa", It may also return "zaz", among other possible answers.
- Assume that:
  - N is an integer within the range [1..1,000];
  - string S consists only of lowercases letters ('a'- 'z') or '?'.
- In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

7. DiversityString

Find a string of a given length containing as many different lower -case letters as possible, in which each letter occurs an equal number of times.

Task description:

- Write a function solution that, given an integer N, returns a string of length N containing as many different lower-case letters ('a' - 'z') as possible, in which each letter occurs an equal number of times.
- Examples:
  - Given N = 3, the function may return "fig", "pea", "nut", etc. Each of these strings contains three different letters with the same number of occurrences.
  - Given N = 5, the function may return "mango", "grape", "melon", etc.
  - Given N = 30, the function may return ""aabbcc..oo" (each letter from 'a' to 'o' occurs twice). The string contains 15 different letters.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..200,000].

8. EraseOneLetter

Remove one letter fromn a word so that resulting string will be as small as possible (alphabetically).

Task description:

- Write a function solution that, given a string S consisting of N characters, returns the alphabetically smallest string that can be obtained by removing exactly one letter from S.
- Examples:
  - Given S= "acb", by removing one letter, you can obtain "ac", "ab" or "cb". Your function should return "ab" (after removing 'c') since it is alphabetically smaller than "ac" and "bc".
  - Given S = "hot", your function should return "ho", which is alphabetically smaller than "ht" and "ot".
  - Given S = "codility", your function should return "cdility", which can be obtained by removing the second letter.
  - Given S = "aaaa", your function should return "aaa". Any occurrence of 'a' can be removed.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [2..100,000];
  - string S is made only of lowercase letters (a-z).

9. EvenPairsOnCycle

Given numbers on a circle, find the maximum number of neighbouring pairs with an even sum.

Task description:

- You are given N numbers on a circle, described by an array A. Find the maximum number of neighbouring pairs whose sums are even. One element can belong to only one pair.
- Write a function: function solution(A); that, given an array A consisting of N integers, returns the maximum number of neighbouring pairs whose sums are even.
- Examples:
  - Given A = [4, 2, 5, 8, 7, 3, 7], the function should return 2. We can create two pairs with even sums: (A[0], A[1] and (A[4], A[5]). Another way to choose two pairs is: (A[0], A[1]) and (A[5], A[6]).
  - Given A = [14, 21, 16, 35, 22],the function should return 1. There is only one qualifying pair: (A[0], A[4]).
  - Given A = [5, 5, 5, 5, 5, 5], the function should return 3. We can create three pairs: (A[0], A[5]), (A[1], A[2]) and (A[3], A[4])
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - each element of array A is an integer within the range [0..1,000,000,000].

10. ForbiddenTriosSwaps

Given a string S consisting of letters 'a' and 'b', return the minimum number of swaps needed to obtain a string with no instances of three identical consecutive letters.

Task description:

- You are given a string S consisting of N letters 'a' and/or 'b'. In one move, you can swap one letter for the other (a' for 'b' or 'b' for 'a').
- Write a function solution that, given such a string S, returns the minimum number of moves required to obtain a string containing no instances of three identical consecutive letters.
- Examples:
  - Given S = "baaaaa", the function should return 1. The string without three identical consecutive letters which can be obtained in one move is "baabaa".
  - Given S = "baaabbaabbba", the function should return 2. There are four valid strings obtainable in two moves: for example, "bbaabbaabbaa".
  - Given S = "baabab", the function should return 0.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [0..200,000];
  - string S is made only of the characters 'a' and/or 'b'.

11. FormatArray

Given an array of integers, output a string that presents the integers in tabular form.

Task description:

- You have an array of numbers and you would like to print these numbers in a tabular format to make them look more organized. Each cell of the table contains exactly one number and is surrounded by exactly four edges:
```
+--+  |4|  +--+ +-----+  |12345|  +-----+
```
- As, you can see above, each corner of the cell is represented by a "+" sign, vertical edges by "-" signs and horizontal edges by "|" signs. The width of the cell adjusts to accommodate the number of digits of the number written within it. There can be many cells in a row. Adjacent cells share an edge:
```
+---+---+---+---+  | 4|  35|  80|123|  +---+---+---+---+
```
- Note that each cell has the same width. The width of the cell adjusts to match the width of the longest number in the table. The numbers in cells are aligned to the right, with any unused area in each cell filled with spaces.
- The table can consist of many rows, and adjacent rows share an edge:
```
+-----+-----+-----+-----+  | 4|  35|  80|  123|  +-----+-----+-----+-----+  |12345|
|  44|  8|  5|  +-----+-----+-----+-----+  | 24|  3|  22|  35|
+-----+-----+-----+-----+
```
- Your goal is to output a table containin all the numbers from a given array such that each row contains exactly K numbers. The last row can contain fewer numbers.
- Write a function: function solution(A, K); that, given a non-empty array A consisting of N integers and an integer K, prints a string representing the formatted aray. The numbers in the table should appear in the same order as the numbers in the array.
- For example, given array A = [4, 35, 80, 123, 12345, 44, 8, 5] and K = 10, the resultant table will contain exactly one row, as shown below:
```
+-----+-----+-----+-----+-----+-----+-----+-----+  | 4|  35|  80|  123|12345|  44|
|  8|  5|  +-----+-----+-----+-----+-----+-----+-----+-----+
```
  - For A = [4, 35, 80, 123, 12345, 44, 8, 5, 24, 3] and K = 4, the table would appear as follows:
```
+-----+-----+-----+-----+  | 4|  35|  80|  123|  +-----+-----+-----+-----+  |12345|
|  44|  8|  5|  +-----+-----+-----+-----+  | 24|  3|  +-----+-----+
```
  - Given A = [4, 35, 80, 123, 12345, 44, 8, 5, 24, 3, 22, 35] and K = 4, the table would appear as follows:
```
+-----+-----+-----+-----+  | 4|  35|  80|  123|  +-----+-----+-----+-----+  |12345|
|  44|  8|  5|  +-----+-----+-----+-----+  | 24|  3|  22|  35|
+-----+-----+-----+-----+
```

Nên dùng set do set là O(1)

(size 2) or a beach layer (size 3).

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

6. SanatoriumAccommodation

Find the minimum number of rooms needed to accommodate all guests in a sanatorium according to their preferences.

Task description:

- There are N guests (numbered from 0 to N-1) in a sanatorium waiting to be assigned a room. In each room, any number of guests can be accommodated. However, not all guests like to have a lot of roommates.
- You are given an array A of N integers: the K-th guest wants to be in a room that contains at most A[K] guests, including themselves.
- Write a function: function solution(A); that, given the array A, returns the minimum number of rooms needed to accommodate all guests.
- Examples:
  - Given A = [1, 1, 1, 1, 1], your function should return 5. Each guest should be accommodated in their own separate room.
  - Given A = [2, 1, 4], your function should return 2. The second guest should be accommodated one room and the other two guests in another room.
  - Given A = [2, 7, 2, 9, 8], your function should return 2. The frst and the third guests should be accommodated in one room and the other three guests in another room.
  - Given A = [7,3, 1, 1, 4, 5, 4, 9], your function should return 4. The guests can be accommodated as follows: the first two guests in one room, the third and the fourth guests in two single rooms, and the other guests in another room.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - each element of array A is an integer within the range [1... .100,000].

7. FixTheTable

Given the distinct positions of holes, find the shortest length of two identical boards that can cover al the holes.

Task description:

- There are N holes arranged in arow in the top of an old table. We want to fix the table by covering the holes with two boards. For technical reasons, the boards need to be of the same length.
- The position of the K-th hole is A[K]. What is the shortest length of the boards required to cover all the holes? The length of the boards has to be a positive integer. A board of length L, set at position X, covers all the holes located between positions X and X+L (inclusive). The position of every hole is unique
- Write a function: function solution(A); which, given an aray A of integers of length N, representing the positions of the holes in the table, returns the shortest board length required to cover all the holes.
- Examples:
  - Given A = [11, 20, 15], your function should return 4. The first board would cover the holes in positions 11 and 15, and the second board the hole at position 20.
  - Given A = [15, 20, 9, 11], your function should return 5. The first board covers the holes at positions 9 and 11, and the second one the holes in positions 15 and 20.
  - Given A = [0, 44, 32, 30, 42, 18, 34, 16, 35], your function should return 18. The first board would cover the holes in positions between 0 and 18, and the second the positions between 30 and 44.
  - Given A = [9], your function should return 1.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - each element of array A is an integer within the range [0..1,000,000,000];
  - the elements of A are all distinct.

8. XYSplit

Given a string, return the number of ways to split it into two parts, such that at least one of the parts has an equal number of 'x's and 'y's.

Task description:

- You are given a string S consisting of N lowercase English letters. In how many ways can we S split into two non-empty parts, such that in at least one part the letter 'x' and the letter 'y' occur the same number of times?
- Write a function: function solution(S); that, given a string S of length N, returns the number of splits S satisfying the condition above.
- Examples:
  - Given S="ayxbx", the function should return 3. There are four possible splits of S: "a/yxbx", "ay/xbx", "ayx/bx" and "ayxb/x". Only "ay/xbx" does not fulfill the condition, so the answer is 3. Note that in "a/yxbx" the left part has 0 occurrences of x' and 'y', So it counts as correct split.
  - Given S = "xzzzy", the function should return 0.
  - Given S = "toyxmy", the function should return 5.
  - Given S = "apple", the function should return 4.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [2..200,000];
  - string S is made only of lowercase letters (a-z).

9. EqualSegments (M)

Find the maximum number of non-intersecting segments of length 2 with equal sums in a given array of integers.

Task description:

- You are given an array A of integers. Find the maximum number of non-intersecting segments of length 2 (two adjacent elements), such that segments have an equal sum.
- For example, given A = [10, 1, 3, 1, 2, 2, 1, 0, 4], there are three non-intersecting segments, each whose sum is equal to 4: (1,3), (2, 2), (0, 4). Another three non-intersecting segments are: (3, 1), (2, 2), (0,4).
- Write a function: function solution(A); that, given an array A of N integers, returns the maximum number of segments with equal sums.
- Examples:
  - Given A = [10, 1, 3, 1, 2, 2, 1, 0, 4], the function should return 3, as explained above.
  - Given A = [5, 3, 1, 3, 2, 3], the function should return 1. Each sum of two adjacent elements is different from the others.
  - Given A = [9, 9, 9, 9, 9], the function should return 2.
  - Given A= [1, 5, 2, 4, 3, 3], the function should return 3. There are three segments: (1, 5), (2, 4), (3, 3) whose sums are equal to 6.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [2..100,000];
  - each element of array A is an integer within the range [0..1,000,000,000].

10. FreeingStorageSpace (M)

Compute how many types of items can stay in a storeroom after removing R consecutive items

Task description:

- A storeroom is used to organize items stored in it on N shelves. Shelves are numbered from 0 to N-1. The K-th shelf is dedicated to items of only one type, denoted by a positive integer A[K].
- Recently it was decided that it is necessary to free R consecutive shelves. Shelves cannot be reordered. What is the maximum number of types of items which still can be stored in the storeroom after freeing R consecutive shelves?
- Write a function: function solution(A, R); that, given an array A of N integers representing types of items stored on storeroom shelves, and an integer R representing the number of consecutive shelves to be freed, returns the maximum number different types of items that can be stored in the storeroom after freeing R consecutive shelves.
- Examples:
  - Given A = [2, 1, 2, 3, 2, 2] and R = 3, your function should return 2. It can be achieved, for example, by freeing shelves 2, 3 and 4 (shelves are numbered from 0).
  - Given A = [2, 3, 1, 1, 2] and R = 2, your function should return 3. All three types can still be stored by freeing the last two shelves.
  - Given A = [20, 10, 10, 10, 30, 20] and R = 3, your function should return 3. It can be achieved by freeing the first three shelves.
  - Given A= [1, 100000, 1] and R = 3, your function should return 0. All shelves need to be freed.
- Write an efficient algorithm for the following asumptions:
  - N is an integer within the range [1..100,000];
  - R is an integer within the range [1..N];
  - each element of array A is an integer within the range [1..100,000].

11. GardenArrangement (M)

Calculate the minimum number of trees that must be planted or replanted in order to make the number of them equal in every section of a garden.

Task description:

- A garden is divided into N sections numbered from 0 to N-1. It is described by an array A, where A[K] denotes the number of trees in the K-th section. To make the garden look more organized, we want the number of trees in every section to be the same. As we don't want to cut any trees down, we can perform either of the following actions:
  - planting a new tree in one of the sections;
  - replanting an existing tree, moving it from one section to another.
- We want to minimize the number of actions performed.
- Write a function: function solution(A); that, given an array A consisting of N integers describing the garden, returns the minimum number of actions we need to perform in order to make all sections of the garden contain the same number of trees.
- Examples:
  - Given A = [1, 2, 2, 4] the function should return 4. We can move one tree from A[3] to A[1] and obtain A = [1, 3, 2, 3]. Then we can plant two trees in A[0] and one tree in A[2] to make every section contain three trees.
  - Given A = [4, 2, 4, 6], the function should return 2. We can move two trees from A[3] to A[1]. This way, every section in the garden will contain four trees.
  - Given A = [1, 1, 2, 1], the function should return 3. We can plant one tree in A[0], A[1] and A[3] so that each section in the garden contains to trees.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - each element of array A is an integer within the range [1..1,000,000,000];
  - the answer is always less than or equal to 2,000,000,000.

12. LongestEvenCount (M)

Given a string, return the longest substring in which every letter occurs an even number of times.

Task description:

- Write a function: function solution(S); that given a string S consisting of N lowercase English letters, returns the length of the longest substring in which every letter occurs an even number of times. A substring is defined as a contiguous segment of a string. If no such substring exists,

- For A = [4, 35, 80, 123, 12345, 44, 8, 5, 24, 3] and K = 4, the table would appear as follows:
  +-----+-----+-----+-----+ | 4| 35| 80| 123| +-----+-----+-----+-----+ |12345| 44| 8| 5| +-----+-----+-----+-----+ | 24| 3| +-----+-----+
- Given A = [4, 35, 80, 123, 12345, 44, 8, 5, 24, 3, 22, 35] and K = 4, the table would appear as follows:
  +-----+-----+-----+-----+ | 4| 35| 80| 123| +-----+-----+-----+-----+ |12345| 44| 8| 5| +-----+-----+-----+-----+ | 24| 3| 22| 35| +-----+-----+-----+-----+
- The function shouldn't return any value.
- You can print a string to the output (without or with the end-of-line character) as follows:
  process.stdout.write('sample string'); process stdout.write('whole line\n');
- Assume that:
  - N is an integer within the range [1..200];
  - K is an integer within the range [1..1,000,000,000];
  - each element of array A is an integer within the range [0..1,000,000, 000):

In your solution, focus on correctness. he performance of your solution will not be the focus of the assessment.

12. SameDigitMerge

Count the number of pairs in which the last digit of the first selected number is the same as the first digit of the second selected number.

Task description:

- There is an array numbers made of N integers. Each number has at least two digits and its first and last digits are different.
- You can select a pair of numbers if the last digit of the first selected number is the same as the first digit of the second selected number. Calculate the number of ways in which such a pair of numbers can be selected.
- Write a function: function solution(numbers); that, given an array numbers made of N integers, returns the number of ways to select a pair of numbers as described above.
- Examples:
  - Given numbers = [30, 12, 29, 91], the function should return 3. The pairs are: (12, 29), (29, 91) and (91, 12).
  - Given numbers = [122, 21, 21, 23], the function should return 5. The pairs are: (122, 21) occuring twice, (122,23), and (21, 122) occurring twice. Please note that the same pair of numbers can appear multiple times if the pairs of their indices are different.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - each element of array numbers is an integer within the range [10..1,000,000,000];
  - the first and last digits in each element of array numbers are different;
  - the answer does not exceed 1,000,000,000.

13. ShortestUniqueSubstring

Given a string, count the length of the shortest word which occurs in it exactly once.

Task description:

- Write a function solution that, given a string S of length N, returns the length of the shortest *unique* substring of S, that is, the length of the shortest word which occurs in S exactly once.
- Examples:
  - Given S = "abaaba", the function should return 2. The shortest unique substring of S is "aa".
  - Given S = "zyzyzyz", the function should return 5. The shortest unique substring of S is "zyzyz". Note that there are shorter words, like "yzy", occurrences of which overlap, but they still count as multiple occurrences.
  - Given S = "aabbbabaaa", the function should return 3. All substrings of size 2 occurs in S at least twice.
- Assume that:
  - N is an integer within the range [1..200];
  - string S is made only of lowercase letters (a-z).

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

14. SmallestDigitSum

Given an integer N, find the smallest integer whose digits sum to N.

Task description:

- Write a function solution that, given integer N returns the smallest non-negative integer whose individual digits sum to N.
- Examples:
  - Given N = 16, the function should return 79. There are many numbers whose digits sum to 16 (for example: 79, 97, 808, 5551, 22822, etc.). The smallest such number is 79.
  - Given N = 19, the function should return 199 (the sum of digits is 1 + 9 + 9= 19).
  - Given N = 7, the function should return 7.
- Assume that:
  - N is an integer within the range [0..50].

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

15. ValueOccurrences

Given a sorted array, calculate how many insertion and removal operations are required to make every value X in the array occur exactly X times.

Task description:

- There is an array A of N integers sorted in non-decreasing order. In one move, you can either remove an integer from A or insert an integer before or after any element of A. The goal is to achieve an array in which all values X that are present in the array occur exactly X times.
- For example, given A = [1, 1, 3, 4, 4, 4], value 1 occurs twice, value 3 occurs once and value 4 occurs three times. You can remove one occurrence each of both 1 and 3, and insert one occurrence 4, resulting in the array [1, 4, 4, 4, 4]. In this array, every element X occurs exactly X times.
- What is the minimum number of moves after which every value X in the array occurs exactly X times?
- Write a function: function solution(A); that, given an array A, returns the minimum number of moves after which every value X in the array occurs exactly X times. Note that it is permissible to remove some values entirely, if appropriate.
- Examples:
  - Given A = [1, 1, 3, 4, 4, 4], your function should return 3, as described above.
  - Given A= [1, 2, 2, 2, 5, 5, 5, 8], your function should return 4. You can delete the 8 and one occurrence of 2, and insert 5 twice, resulting in [1, 2, 2, 5, 5, 5, 5, 5] affer four moves. Notice that after the removals, there is no occurrence of 8 in the array anymore.
  - Given A = [1, 1, 1, 1, 3, 3, 4, 4, 4, 4, 4], your function should return 5.
  - Given A = [10, 10, 10], your function should retum 3. You can remove all elements, resulting in an empty array.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - each element of array A is an integer within the range [1..100,000,000];
  - elements of array A are sorted in non-decreasing order.

16. AsphaltPatches

Given a description of a road with potholes, find the minimum number of patches needed to repair them all. One patch repairs three consecutive road segments.

Task description:

- There is a road consisting of N segments, numbered from 0 to N-1, represented by a string S. Segment S[K] of the road may contain a pothole, denoted by a single uppercase "x" character, or may be a good segment without any potholes, denoted by a single dot, ".".
- For example, string ".x..x" means that there are two potholes in total in the road: one is located in segement S[1] and one in segment S[4]. All other segments are good.
- The road fixing machine can patch over three consecutive segments at once with asphalt and repair all the potholes located within each of these segments. Good or already repaired segments remain good after patching them.
- Your task is compute the minimum number of patches required to repair all the potholes in the road.
- Write a function: function solution(S); that, given a string S of length N, returns the minimum number of patches required to repair all the potholes.
- Examples:
  - Given S = ".x..x", your function should return 2. The road fixing machine could patch, for example, segments 0–2 and 2–4.
  - Given S = "x.xxxxx.x.", your function should return 3. The road fixing machine could patch, for example, segments 0–2, 3–5 and 6–8.
  - Given S = "xx.xxx.", your function should return 2. The road fixing machine could patch, for example, segments 0–2 and 3–5.
  - Given S = "xxxx", your function should return 2. The road fixing machine could patch, for example, segments 0–2 and 1–3.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [3.. 100,000];
  - string S is made only of the characters '.' and/or 'x'.

17. EndsTheSame (M)

Count how many times the first and last letters of a string are the same during a sequence of operations.

Task description:

- Initially, string S of length N is given. Then N-1 operations are applied to it: move the first letter of S to the end. How many times is the first letter of S the same as the last letter
- For example, given S = "abbaa", the obtained sequence of strings is: abbaa - bbaaa --> baaab --> aaabb - aabba. Three of them have the same first and last letter.
- Write a function: function solution(S); that,given a string S of length N, consisting of lettes 'a' and/or 'b', returns the number of times the first letter is the same as the last in the obtained sequence of strings.
- Examples:
  - Given S = "abbaa" the function should return 3, as described above.
  - Given S = "aaaa", the function should return 4. The first and last letters are always the same.
  - Given 'S= "abab", the function should return 0. The first and last letters are always different.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [2..200,000];
  - string S is made only of the characters 'a' and/or 'b'.

18. ABString

Check, whether in a given string all letters 'a' occur before all letters 'b'.

Task description:

- Write a function solution that, given a string S consisting of N letters 'a' and/or 'b' returns true when all occurrences of letter 'a' are before all occurrences of letter 'b' and returns false otherwise.
- Examples:
  - Given S = "aabbb", the function should return true.

Nên dùng set do set là O(1)

Done 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18

Array
Linked list (singly)
Stack (Operations O(1))
Queue (Operations O(1))
Set
Hash Table

Dfs, bfs, binary search

Sliding window, 2 pointers

Note: Coi kỹ map & set, queue

Giả sử s = "abcd" và t = "abcde";

Sau bước 2, HashMap sẽ chứa: {'a': 1, 'b': 1, 'c': 1, 'd': 1, 'e': 1}.
Sau bước 3, HashMap sẽ chứa: {'e': 1} (vì các ký tự 'a', 'b', 'c', 'd' đã bị loại bỏ).
Kết quả trả về sẽ là 'e'.

Giả sử nums = [2, 7, 11, 15] và target = 9:

Khi i = 0, nums[i] = 2, res = 9 - 2 = 7. Map không có 7, nên thêm 2 vào Map với vị trí 0.
Khi i = 1, nums[i] = 7, res = 9 - 7 = 2. Map có 2, nên trả về [map.get(2), 1] tức là [0, 1].

Làm sao để phân biệt Palindrome của chuỗi và số đối xứng

12. LongestEvenCount (M)

Given a string, return the longest substring in which every letter occurs an even number of times.

Task description:

- Write a function: function solution(S); that given a string S consisting of N lowercase English letters, returns the length of the longest substring in which every letter occurs an even number of times. A substring is defined as a contiguous segment of a string. If no such substring exists, return 0.
- Examples:
  - Given S= "bdaaadadb", the function should return 6. Substrings in which every letter occurs an even number of times are "aa", "adad", "daaada" and "aaadad". The length of the longest of them is 6.
  - Given S = "abacb", the function should return 0. There is no non - empty substring in which every letter occurs an even number of times.
  - Given S = "zthtzh", the function should return 6. Every letter in the whole string occurs an even number of times.
- Write an efficient algorithm for the following assumptions:
  - N an integer within the range [1..100,000];
  - string S consists only of lowercase letters ('a' - 'z').

13. ArraySlicing (M)

Given a sequence of distinct integers, split it into the maximum possible number of slices such that sorting the separate slices sorts the full sequence.

Task description:

- We are given an array A consisting of N distinct integers. We would like to sort array A into ascending order using a simple algorithm. First, we divide it into one or more slices (a slice is a contiguous subarray). Then we sort each slice. After that, we join the sorted slices in the same order.
- Write a function solution that returns the maximum number of slices for which the algorithm will a return a correctly sorted array.
- Examples:
  - Given A = [2, 4, 1, 6, 5, 9, 7], the function should return 3. The array can be split into three slices: [2, 4, 1], [6, 5] and [9, 7]. Then, after sorting each slice and joining them together, the whole array will be sorted into ascending order.
  - Given A = [4, 3, 2, 6, 1], the function should return 1. The array cannot be split into smaller slices; it has to be sorted all at once.
  - Given A = [2, 1, 6, 4, 3, 7], the function should return 3
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - each element of array A is an integer within the range [1..1,000,000,000];
  - the elements of A are all distinct.

14. LongestTwoDigitFragment (M)

Find the length of the longest fragment in a given array which can be written down using at most two different digits.

Task description:

- Write a function: function solution(A); that, given an array A of N integers, returns the length of the longest consistent fragment of A in which all elements can be generated using at most two different digits. You must use the sam digits for all elements.
- Examples:
  - Given A = [23, 333, 33, 30, 0, 505], the function should return 4. Elements 333, 33, 30 and 0 can be generated using only digits 0 and 3.
  - Given A = [615, 88, 498, 99, 9], the function should return 2. The last two elements can be generated using only digit 9.
  - Given A =[123, 456], the function should return 0.
- Assume that:
  - N is an integer within the range [1..100];
  - each element of array A is an integer within the range [1..1,000, ,000,000].

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

15. MinDistinct (M)

Given an array of N numbers within the range [1..N], find the minimum number of increments and decrements of a single element required to make the array a permutation.

Task description:

- You are given an array A consisting of N integers within the range [1..N]. In one move, you can increase or decrease the value of any element by 1. After each move. all numbers should remain within the range [1..N].
- Your task is to find the smallest required number of moves to make all elements in the array pairwise distinct (in other words, no value can appear in the array more than once).
- Write a function: function solution(A); that, given an array A consisting of N integers, returns the smallest number of moves required to make all elements in the array pairwise distinct. If the result is greater than 1,000,000,000, the function should return -1.
- Examples:
  - Given A = [1, 2, 1], the function should return 2, because you can increase A[2] twice: [1, 2, 1] --> [1, 2, 2] --> [1, 2, 3]. In this example, you could also change the array to the following values in two moves: [3, 2, 1], [1, 3, 2], [2, 3, 1].
  - Given A = [2, 1, 4, 4], the function should return 1, as it is sufficient to decrease A[2] or A [3] by 1, resulting in [2, 1, 3, 4] or [2, 1, 4, 3].
  - Given A = [6, 2, 3, 5, 6, 3], the function should return 4, because you can achicve the following array in four moves: [6, 2, 1, 5, 4, 3].
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1 ..200,000];
  - each element of array A is an integer within the range [1..N].

16. PathDetection (M)

Check whether the given graph contains a Path going through all vertices,one by one, in increasing order of their numbers.

Task description:

- You are given an undirected graph consisting of N vertices, numbered from 1 to N, and M edges.
- The graph is described by two arays, A and B, both of length M. A pair (A[K], B[K]), for K from 0 to M-1, describes an edge between vertex A[K] and vertex B[K].
- Your task is to check whether the given graph contains a path from vertex 1 to vertex N going through all of the vertices, one by one, in increasing order of their numbers. All connections on the path should be direct.
- Write a function: function solution(N, A, B); that, given an integer N and two arrays A and B of M integers each, returns true if there exists a path from vertex 1 to N going through all vertices, one by one, in increasing order, or false otherwise.
- Examples:
  - Given N = 4, A = [1, 2, 4, 4, 3] and B = [2, 3, 1, 3, 1], the function should return true. There is a path (1 --> 2 - 3 --> 4) using edges (1, 2), (2, 3) and (4, 3).
  - Given N = 4, A = [1, 2, 1, 3] and B = [2, 4, 3, 4], the function should return false. There is no path (1 -> 2 --> 3 --> 4), as there is no direct connection from vertex 2 to vertex 3.
  - Given N = 6, A = [2, 4, 5, 3] and B = [3, 5, 6, 4], the function should return false. There is no direct connection from vertex 1 to vertex 2.
  - Given N = 3, A = [1, 3] and B =[2, 2], the function should return true. There is a path (1 --> 2 --> 3) using edges (1, 2) and (3, 2).
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [2..100,000];
  - M is an integer within the range [0..100,000];
  - all elements of arrays A and B are integers within the range [1..N];
  - there are no self-loops (edges with A[K] = B[K]) in the graph;
  - there are no multiple edges between the same vertices.

17. PriceFluctuation (M)

For a historical record of assets values, calculate the maximum income you could have made.

Task description:

- You are give a record of the historical prices of an investment asset from the last N days. Analyze the record in order to calculate what could have been your maximum income. Assume you started with one asset of this type and could hold at most one at a time. You could choose to sell the asset whenever you held one. If you did not hold an asset at some moment, you could always afford to buy an asset (assume you had infinite money available).
- What is the maximum income you could make?
- Write a function: function solution (A); that, given en array A of length N representing a record of prices over the last N days, returns the maximum income you could make. As the result may be large, return its last nine digits without leeding zeros (return the result modulo 1,000,000,000).
- Examples:
  - Given A = [4, 1, 2, 3], the function should return 6. You could sell the product on the first day (for 4), buy it on the second (for 1) and sell it again on the last day (for 3). The income would be equal 4 - 1 + 3 = 6.
  - Given A= [1, 2, 3, 3, 2, 1, 5], the function should return 7. You could sell the product when its value was 3, buy it when it changed to 1, and sell it again when it was worth 5.
  - Given A = [1000000000, 1, 2, 2, 1000000000, 1, 1000000000], the function should return 999999998. The maximum possible income is 2999999998, whose last 9 digits are 999999998.
- Write an efficiency algorithm for the following assumptions:
  - N is an integer within the range [1..200,000];
  - each element of array A is an inlegor within the range [0..1,000,00,000].

18. Sticks (M)

Given two sticks, cut them into four sticks in order to create the largest possible square.

Task description:

- There are two wooden sticks of lengths A and B respectively. Each of them can be cut into shorter sticks of integer lengths. Our goal is to construct the largest possible square. In order to do this, we want to cut the sticks in such a way as to achieve four sticks of the same length (note that there can be some leftover pieces). What is the longest side of square that we can achieve?
- Write the function: function solution(A, B); that given two integers A, B, returns the side length of the largest square that we can obtain. If it is not possible to create any square, the function should return 0.
- Examples:
  - Given A = 10, B = 21, the function should return 7. We can split the second stick into three sticks of length 7 and shorten the first stick by 3.
  - Given A = 13, B = 11, the function should return 5. We can cut two sticks of length 5 from each of the given sticks.

```
public class LCM {
    public static void main(String[] args) {
        int a = 95;
        int b = 55;
        int lcm = findLCM(a, b);
        System.out.println("Bội chung nhỏ nhất của " + a + " và " + b + " là: " + lcm);
    }

    public static int findLCM(int a, int b) {
        return (a * b) / findGCD(a, b);
    }

    public static int findGCD(int a, int b) {
        if (b == 0) {
            return a;
        }
        return findGCD(b, a % b);
    }
}
```

Liên tục chia số lớn hơn cho số nhỏ hơn và lấy phần dư cho đến khi phần dư bằng 0. Khi đó, số chia cuối cùng là UCLN.

Vì thuật toán Euclid thực hiện các phép chia và lấy phần dư, mỗi phép toán này có độ phức tạp là (O(1)).
Tổng số lần lặp lại là (O(\log(\min(a, b)))), vì giá trị nhỏ hơn giữa a và b sẽ quyết định số lần lặp lại.

Functional Testing

Functional testing verifies that software operates according to specified requirements, focusing on user interactions, features, and business logic.

1. **Unit Testing**:
 - Tests individual components for correctness.
 - Typically automated and conducted by developers.

2. **Integration Testing**:
 - Ensures modules or services work together.
 - Identifies interface defects between units.

3. **System Testing**:
 - Evaluates the complete software system.
 - Checks end-to-end specifications.

4. **User Acceptance Testing (UAT)**:
 - Validates the software against business needs

18. ABString

Check, whether in a given string all letters 'a' occur before all letters 'b'.

Task description:

- Write a function solution that, given a string S consisting of N letters 'a' and/or 'b' returns true when all occurrences of letter 'a' are before all occurrences of letter 'b' and returns false otherwise.
- Examples:
  - Given S = "aabbb", the function should return true.
  - Given S = "ba", the function should return false.
  - Given S = "aaa", the function should return true. Note that 'b' does not need to occur in S.
  - Given S = "b", the function should return true. Note that 'a' does not need to occur in S.
  - Given S = "abba", the function should return false.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..300,000];
  - string S is made only of the characters 'a' and/or 'b'.

19. MonitorsDelivery

Given a set of orders, calculate how many of them can be fulfilled.

Task description:

- A technology company announced that a new supply of P monitors would soon be available at their store. There were N orders (numbered from 0 to N-1) placed by customers who wanted to buy those monitors. The K-th order has to be delivered to a location at distance D[K] from the store and is for exactly C[K] monitors.
- Now the time has come for the monitors to be delivered. The orders will be fulfilled one by one. To minimize the shipping time, it has been decided that the deliveries will be made in order of increasing distance from the store, If there are many customers at the same distance, they can be processed in any order. Monitors to more distant customers will be delivered only once all orders to customers closer to the store have already been fulfilled.
- What is the maximum total number of orders that can be fulfilled?
- Write a function: function solution(D, C, P); that, given two arrays of integers D and C and an integer P, returns the maximum total number of orders that can be fulfilled.
- Examples:
  - Given D = [5, 11, 1, 3], C= [6, 1, 3, 2] and P = 7, the function should return 2. The customers at distances 1 and 3 will have their orders fulfilled and 3 + 2 = 5 monitors will be delivered.
  - Given D = [10, 15, 1], C= [10, 1, 2] and P= 3, the function should return 1. Only the order for the customer at distance 1 will be fulfilled. There will not be enough monitors in the store for the customer at distance 10. Therefore, orders for customers at distances 10 and 15 will not be fulfilled.
  - Given D = [11, 18, 1], C = [9, 18, 8] and P= 7, the function should return 0.
  - Given D = [1, 4, 2, 5], C = [4, 9, 2, 3] and P = 19, the function should return 4.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..100,000];
  - each element of arrays D and C is an integer within the range [1..1,000,000,000];
  - P is an integer within the range [0..1,000,000,000].

Automation:
Java:
Class vs Object
Method vs Function
OOP
Abstract và Interface
Constructor
Access Modifier
Biến static
Collections java
Overload and override

Selenium:
Lý thuyết Selenium/Appium là gì
Các loại wait
Handle Iframe / Window handle
Lấy Xpath cho Element
Tương tác và verify Element: Clickable/ Enabled/ Visibled/ Upload file

Advance: các design pattern: POM, singleton, factory, Selenium grid, Browser stack
Cmd:
Run project by command line
Setup to run 5/100 case by command line

SQL:

Các câu lệnh CRUD cơ bản
Các loại Join
Distinct/Group by

API:

GET / POST/ PUT/ PATCH/ DELETE
STUB and MOCK
Authentication vs Authorization
Các mã lỗi cơ bản (ví dụ cụ thể)
Mã 409
Header/Body cần lưu ý hay test gì
Lí thuyết về contract testing
Thập Pyramid
Trong test Api, phần nào là quan trọng nhất
Postman vs Newman

Agile working process:

Các main event
Working process ở dự án hiện tại
event nào quan trọng nhất
Sprint plan ra sao

Manual Testing:

Test Level
Test Technical
Retest vs Regression Test
Handle Bug Lifecycle
Quản lý task/bug/req
Cách viết TCs
Các thành phần quan trong trong TCs
Độ phức tạp/ Độ ưu tiên của TCs/Bug
Test Plan/ Test Report nếu từng làm
Khi test webui và test mobile thì thấy có những khác nhau gì
Flaky test là gì

CI/CD:

Setup CI/CD cho dự án chưa?
Dùng Jenkins hay gì
Type of project trên Jenkins
Muốn setup run daily lúc 12h đêm thì làm tn
Muốn email/ send report sau khi run xong thì tn

GIT:

Các câu lệnh GIT cơ bản
Giải quyết git conflict
Git rebase, cherry pick
...

square that we can achieve?

- Write the function: function solution(A, B); that given two integers A, B, returns the side length of the largest square that we can obtain. If it is not possible to create any square, the function should return 0.
- Examples:
  - Given A = 10, B = 21, the function should return 7. We can split the second stick into three sticks of length 7 and shorten the first stick by 3.
  - Given A = 13, B = 11, the function should return 5. We can cut two sticks of length 5 from each of the given sticks.
  - Given A = 2, B = 1, the function should return 0. It is not possible to make any square from the given sticks.
  - Given A = 1, B = 8, the function should return 2. We can cut stick B into four parts.
- Write an efficient algorithm for the following assumptions:
  - A and B are integers within the range [1..1,000,000,000].

19. AngryFrogs (M)

Given blocks in a row, find the distance between the two most distant blocks, such that the heights between them are first decreasing and then increasing.

Task description:

- There N blocks, numbered from 0 to N-1, arranged in a row. A couple of frogs were sitting together on one block when thay had a terrible quarrel. Now thay want to jump away from one another so that the distance between them will be as large as possible. The distance between blocks numbered J and K, where J <= K, is computed as K - J + 1. The frogs can only jump up, meaning that thay can move from one block to another only if the two blocks are adjacent and the second block is of the same or greater height as the first. What is the longest distance that thay can possibly create between each other, if they also chose to sit on optimal starting block initially?
- Write a function: function solution(blocks); that, given an array blocks consiting of N integers denoting the heights of the blocks, return the longest possible distance that two frogs can make between each other starting from one of the blocks.
- Examples:
  - Given blocks = [2, 6, 8, 5], the function should return 3. If starting from blocks[0], the first frog can stay where it is and the second frog can jump to blocks[2] (but not to blocks[3]).
  - Given blocks = [1, 5, 5, 2, 6], the function should return 4. If starting from blocks[3], the first frog can jump to blocks[1], but not blocks[0], and the second frog can jump to blocks[4].
  - Given blocks = [1, 1], the function should return 2. If starting from blocks[1], the first frog can jump to blocks[0] and the second frog can stay where it is. Starting from blocks[0] would result in the same distance.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [2..200,000];
  - each element of array blocks is an integer within the range [1..1,000,000,000].

20. AssassinsStealth (M)

Given a two-dimensional board with an assassin and guards on it, determine whether the assassin can get to the bottom-right corner undetected.

Task descriptions:

- We are given a two-dimensional board of size N x M (N rows and M columns). Each field of the board can be empty ('.'), may contain an obstacle ('X') or may have a character in it. The character might be either an assassin ('A') or a guard. Each guard stands still and looks straight ahead, in the direction they are facing.
- Every guard looks in one of four directions (up, down, left or right on the board) and is represented by one of four symbols. A guard denoted by '<' is looking to the left; by '>', to the right; '^', up; or 'v', down. The guards can see everything in a straight line in the direction in which they are facing, as far as the first obstacle ('X' or any other guard) or the edge of the board.
- The assassin can move from the current field to any other empty field with a shared edge. The assassin cannot move onto fields containing obstacles or enemies.
- Write a function: function solution(B); that, given an array B consiting of N strings denoting rows of the array, return true if is it possible for the assassin to sneak from their current location to the bottom-right cell of the board undetected, and false otherwise.
- Examples:
  - Given B = ["X....>", ".v..X.", ".>..X.", "A....."], your function should return false. All available paths lead through a field observed by a guard.
  - Given B = ["...Xv", "AX...", ".XX..."], your function should return true.The guard in the second row is blocking the other one from watching the bottom-right square.
  - Given B = ["...", ".<.A"], your function should return false, as the assassin gets spotted right at the start.
  - Given B = ["A.v", "..."], your function should return false. It's not possible for the assassin to enter the bottom -right cell undetected, as the cell is observed.
- Write an efficient algorithm for the following assumptions:
  - N is an integer within the range [1..500];
  - all strings in B are of the same length M from range [1..500];
  - there is exactly one assassin on the board;
  - there is no guard or wall on B[N-1][M-1];
  - every string in B consits only of the following characters '.', 'x', '<', '>', 'v', '^' and/ or 'A'.

21. ChoosingNumbers (M)

Choose the maximum number of integers from an array so that, after sorting, the difference between every adjacent pair is equal.

Task description:

- There is an array A made of N integers. Your task is to choose as many integers from A as possible so that, when they are put in ascending order, all of the differences between all pairs of consecutive integers are equal.
- For example, for A = [4, 3, 5, 1, 4, 4], you could choose 1, 3 and 5 (with differences equal to 2) or 4, 4 and 4 (with differences equal to 0).
- What is the maximum number of integers that can be chosen?
- Write a function: function solution(A); that, given an array A made of N integers, return the maximum number of integers that can be chosen following the rules described abow.
- Examples:
  - A = [4, 7, 1, 5, 3], the function should return 4. It is possible to choose four integers (7, 1, 5 and 3). When put in ascending order, the difference between all consecutive integers is 2.
  - A = [12, 12, 12, 15, 10], the function should return 3. It is optimal to choose all integers with a value 12.
  - A = [18, 26, 18, 24, 24, 20, 22], the function should return 5. Five integers (18, 20, 22, 24, 26) can be chosen. Notice that we cannot pick any other integers, even though they occur more than once.
- Assume that:
  - N is an integer within the range [2..50];
  - each element of an array is an integer within the range [1..100].

22. CleaningRobot

Count the number of squares on a rectangulae grid visited by a cleaning robot. The robot only moves forward, turning right if it cannot make a move.

Task description:

- There is a cleaning robot which is cleaning a rectangular grid of size NxM, representted by array R consiting of N strings. Rows are numbered from 0 to N-1 (from top to bottom) and columns are numbered from 0 to M-1 (from left to right).
- The robot starts cleaning in the top-left corner, facing rightwards. It moves in a straight line for as long as it can, in other words, while there is an unoccupied grid square ahead of it. When it cannot move forward, it rotates 90 degrees clockwise and tries to move forward again until it encounters another obstacle, and so on. Dots in array ('.') represent empty squares and "x"s represent occupied squares (ones the robot cannot move through). Each square that the robot occupied at least once is considered clean. The robot moves idenfinitely.
- Write a function: function solution(R); that, given an array R consiting of N strings, each of length M, representing the grid, returns the number of clean squares.
- Examples:
  - A ["...x..", "....xx", "..x..."], your function should return 6. The robot starts at (0, 0), facing rightwards, and moves to (0, 2), where it turn due to the obstacle at (0, 3). Then it goes down from (0, 2) to (1, 2), where it changes direction again due to another obstacle. Next it goes left from (1, 2) to (1, 0), where it turns once because of the grid boundary, then it moves once and turns once more, which make it stand again at position (0, 0) facing rightwards, just as at the beginning, which means it will now repeat the loop indefinitely. The total number of cleaned squares is 6.
  - A = ["...x..", "x.....", "....x.", "....."], your function should return 15.
  - A = ["...x.", ".x..x", "x..x", ".x.."], your function should return 9.
  - A = ["."], your function should return 1, because there is only one square on the grid and it is cleaned in the first move.
- Assume that:
  - N and M are integers within the range [1..20];
  - Top-left cell is empty;
  - each string in R consists only of the following characters "." and/or "x".

23. CollectingRainwater

Find the minimum number of water tanks needed to save the rainwater from all houses in a street.

Task description:

- Today our world approaching an ecological crisis. Due to global warning, the sea level is rising. At the same time, the amount of drinkable water is decreasing. One idea about preventing the loss of drinkable water is the propagation of rainwater storage, in other words, equipping houses with a water tank for rainwater.
- You are given a string S describing a street, in which 'H' denotes a house and '-' denotes an empty plot. You may place water tanks in empty plots to collect rainwater from nearby houses. A house can collect its own rainwater if there is a tank next to it (on either the left or the right side).
- Your task is to find the minimum number of water tanks needed to collect rainwater from all of the houses.
- For example, given S = "-H-HH---", you can collect rainwater from all three houses by using two water tanks. You can position one water tank between the first and second houses and the other after the third house. This placement of water tanks can be represented as "-HTHHT-", where "T" denotes a water tank.
- Write a functions: function solution(S); that, given a string S of length N, returns the minimum number of water tanks needed.
- If there is no solution, return -1;
- Examples:
  - S = "-H-HH---", the function should return 2, as explained above.

2. **Integration Testing**:
 - Ensures modules or services work together.
 - Identifies interface defects between units.

3. **System Testing**:
 - Evaluates the complete software system.
 - Checks end-to-end specifications.

4. **User Acceptance Testing (UAT)**:
 - Validates the software against business needs.
 - Conducted by end-users or clients.

5. **Smoke Testing**:
 - Basic tests to check critical functionalities.
 - Determines build stability for further testing.

6. **Sanity Testing**:
 - Focuses on specific functions or bug fixes.
 - Ensures rational behavior post-fixes.

7. **Regression Testing**:
 - Confirms new changes haven't affected existing features.
 - Often automated for efficiency.

Non-Functional Testing

Non-functional testing evaluates aspects like performance, usability, and reliability.

1. **Performance Testing**:
 - Assesses speed, scalability, and stability.
 - Includes load, stress, and endurance testing.

2. **Security Testing**:
 - Identifies vulnerabilities and ensures data protection.
 - Includes penetration testing and risk assessment.

3. **Usability Testing**:
 - Evaluates user-friendliness and interface design.
 - Focuses on user experience and accessibility.

4. **Compatibility Testing**:
 - Verifies software performance across environments.
 - Includes browser, OS, and device compatibility.

5. **Reliability Testing**:
 - Ensures consistent performance over time.
 - Involves failure rate and recovery testing.

6. **Compliance Testing**:
 - Checks adherence to standards and regulations.
 - Ensures legal and industry compliance.

7. **Localization Testing**:
 - Verifies software for different regions and languages.
 - Ensures cultural and linguistic accuracy.

    - S = "H", the function should return -1. There is no available plot on which to place a water tank.
    - S = "HH-HH", the function should return -1. There is only one plot to put a water tank, and it is impossible to collect rainwater from the first and last houses.
    - S = "-H-H-H-H-H", the function should return 3. One possible way of placing water tank is "-HTH-HTHTH".
- Assume that:
    - N is an integer within the range [1..20];
    - string S is made only of the characters '-' and/or 'H'.

24. CreateDiverseWord

Join some of the given strings to create the longest possible string that contains neither 'AAA' nor 'BBB' as fragment.

Task description:

- There are two-letter strings, "AA", "AB" and "BB", which appear AA, AB and BB times respectively. The task is to join some of these strings to create the longest possible string which does not contain "AAA" or "aaa".
- For example, having AA = 5, AB = 0 and BB = 2, it is possible to join five string by taking both of the "AB" strings and three of the "AA" strings. Then they can be joined into "AA-BB-AA-BB-AA" -> "AABBAABBAA".
- Note that if it is not possible to add another "AA" string as the result would then contain "AAA".
- Write a function: function solution(AA, AB, BB); that, given three integers AA, AB and BB, returns the longest string that canbe created according to the rules described above. If there is more than one possible answer, the function may return any of them.
- Examples:
    - AA = 5, AB = 0 and BB = 2, the function should return "AABBAABBAA", as explained above.
    - Given AA = 1, AB = 2 and BB = 1, possible result are "BBABABAA", "ABAABBAB", "ABABAABB" or "AABBABAB"
    - Given AA = 0, AB = 2 and BB = 0, the function should return "ABAB"
    - Given AA = 0, AB = 0 and BB = 10, the function should return "BB"
- Assume that:
    - AA, AB and BB are integers within the range [0..10];
    - the resulting string will not be empty.

25. DivedeIntoGroups

Divide people into three groups so that the largest score difference between the people in each group is as small as possible.

Task description:

- There is an array A consiting of N integers. Divide them into three non-empty groups. In each group we calculate the difference between the largest and smallest integer. Our goal is to make the maximum of these differences as small as possible.
- For example, given A = [11, 5, 3, 12, 6, 8, 1, 7, 4], we can divide the element into three groups:
    - [3, 1, 4] - the difference between elements is 3;
    - [5, 6, 7, 8] - the difference is also 3;
    - [11, 12] - the difference is 1.
- The maximum difference equals 3, which is the minimum possible result.
- Write a function: function solution(A); that, given an array A. returns the minimum possible result as explained above.
- Examples:
    - For A = [11, 5, 3, 12, 6, 8, 7, 4], the function should return 3, as explained above.
    - For A = [10, 14, 12, 1000, 11, 15, 13, 1], the function should return 5. The elements of A should be divided into three groups as follows:
        - [1];
        - [10, 14, 12, 11, 15, 13];
        - [1000].
    - For A = [4, 5, 7, 10, 10, 12, 12, 12], the function should return 2. The elements of A should be divided into three groups as follows:
        - [4, 5];
        - [7];
        - [10, 10, 12, 12].
    - For A = [5, 10, 10, 5, 5], the function should return 0. The first group may contain all elements with value 5; the second and the third groups may each contain one element with value 10.
- Write an efficient algorithm for the following assumptions:
    - N is an integer within the range [3..100,000];
    - each element of array A is an integer within the range [0..1,000,000,000].