Amazon Cracker Sheet

Note: [To Use this sheet optimally go and watch the "Cracking the Amazon coding interview🔥: The definitive prep guide" video on Debug Buzz Channel.
Link - https://youtu.be/BEj_Oues8lE]

Q) 9th August 2022

1. Code Question 1

Given an integer denoting a total number of wheels, help Amazon Logistics find the number of different ways to choose a fleet of vehicles from an infinite supply of two-wheeled and four-wheeled vehicles such that the group of chosen vehicles has that exact total number of wheels. Two ways of choosing vehicles are considered to be different if and only if they contain different numbers of two-wheeled or four-wheeled vehicles.

For example, if our array *wheels = [4,5,6]* our return array would be *res = [2, 0, 2]*. Case by case, we can have *1* four-wheel or *2* two-wheel to have *4* wheels. We cannot have *5* wheels. We can have *1* four-wheel and *1* two-wheel or *3* two-wheel vehicles in the final case.

Function Description
Complete the function *chooseFleets* in the editor below. The function should return an array of integers representing the answer for each *wheels[i]*.

chooseFleets has the following parameter(s):
   wheels[wheels[0],...wheels[n-1]]: an array of integers

Constraints

- $1 \le n \le 10^5$
- $1 \le wheels[i] \le 10^6$

▸ **Input Format for Custom Testing**

▾ **Sample Case 0**

Sample Input 0

```
3
6
3
2
```

Your team at Amazon is working on a system that divides applications to a mixed cluster of computing devices. Each application is identified by an integer ID, requires a fixed non-zero amount of memory to execute, and is defined to be either a foreground or background application. IDs are guaranteed to be unique within their own application type, but not across types.

Each device should be assigned two applications at once, one foreground application and one background application. Devices have limited amounts of memory and cannot execute applications that require more memory than the available memory. The goal of the system is to maximize the total utilization of the memory of a given device. A foreground/background application pair is considered to be optimal if there does not exist another pair that uses more memory than this pair, and also has a total less than or equal to the total memory of the device. For example, if the device has 10MB memory, a foreground/background pair using a sum total of 9MB memory would be optimal if there does not exist a pair that uses a sum total of 10 MB, but would not be optimal if such a pair did exist.

Write an algorithm to find the sets of foreground and background application pairs that optimally utilize the given device for a given list of foreground applications and a given list of background applications.

Input
The input to the function/method consists of three arguments:
deviceCapacity, an integer representing the maximum capacity of the given device;
foregroundAppList, a list of pairs of integers where the first integer represents the unique ID of a foreground application and the second integer represents the amount of memory required by this application;
backgroundAppList, a list of pairs of integers where the first integer represents the unique ID of a background application and the second integer represents the amount of memory required by this application.

Output
Return a list of pairs of integers representing the pairs of IDs of foreground and background applications that optimally utilize the given device [foregroundAppID,backgroundAppID]. If no pair is possible, return a list with empty pair - not just an empty list.

Examples

Example 1:
Input:
deviceCapacity = 7
foregroundAppList = [[1,2],[2,4],[3,6]]
backgroundAppList = [[1,2]]

Output:
[[2,1]]

Explanation:

Q) 8th August 2022

You are given an array A of N integers. You need to find two integers x and y such that the sum of the absolute difference between each element of the array to one of the two chosen integers is minimal.

Task

Determine the minimum value of the expression $\sum_{i=1}^{n} \min(abs(a[i]-x), abs(a[i]-y))$ if the chosen numbers are x and y.

Example1:

N = 4

A = [2, 3, 6, 7]

Approach

You can choose the two integers, 3 and 7.

The required sum = |2 - 3| + |3 - 3| + |6 - 7| + |7 - 7| = 1 + 0 + 0 + 1 = 2.

Example2:

Given

N = 3

A = [1, 3, 5]

Approach

You can choose the two integers, 1 and 4.

The required sum = |1 - 1| + |3 - 4| + |5 - 4| = 0 + 1 + 1 = 2.

The second test case

Example3:

Given

N = 4

A = [3, 2, 5, 11]

Approach

You can choose the two integers, 3 and 11.

The required sum = |2 - 3| + |3 - 3| + |5 - 3| + |11 - 11 |= 1 + 0 + 2 + 0 = 3.

Q) 8th August 2022

A Z sequence is defined as:

$Z_i = P \times X(Z_{i-1}) + Q$ for i>0

Z_i = 2 for i=0

X(K) is defined as the number of set bits in the binary form of a number K.

Print the number of set bits in the binary form of ZN

Example

N = 2

P = 1

Q = 3

Approach

So , Z[0] = 2 , Z[1] = P$X[2]$+Q. Now X[2] = 1 as 2 can be written as 10 in binary form So, Z[1] = 11+3 = 4 . Similarly , Z[2] = P$X[4]$+Q .Now 4 can be written as 100 . So , Z[2] = 11+3 = 4. Now answer is the number of set bits in Z[2] = 4, so 1.

EXAMPLE:

Sample input

1

3 5 1

Sample output

1

Explanation

Based on expression, sequence will be like 2, 8, 8,...

As N=1 so Z1 is 8 Number of set bits in 8(i.e 1000) is 1

Q) 7th August 2022

Suppose you are given a grid of 1's and 0's. All adjacent 1's are connected components.

For example, in the following case you have 2 connected components because you have two "islands" of 1's.

```
1 1 0 0 1 1
1 0 0 0 1 1
1 0 0 0 0 0
```

Now you have a function called insertValue(coordinates) which takes in a row and column and inserts a 1. The function must return the updated number of connected components. So for example:

init:

```
1 1 0 0 1 1
1 0 0 0 1 1
1 0 0 0 0 0
```

insertValue(row=1, col=1) gives 2 connected components still because

grid is:

```
1 1 0 0 1 1
1 1 0 0 1 1
1 0 0 0 0 0
```

insertValue(row=1, col=2) gives 2 connected components still because

grid is:

```
1 1 0 0 1 1
1 1 1 0 1 1
1 0 0 0 0 0
```

insertValue(row=1, col=3) gives 3 connected components still because

grid is:

```
1 1 0 0 1 1
1 1 1 1 1 1
1 0 0 0 0 0
```

Q) 6th August 2022

1	2	3
abc	def	ghi
4 jkl	5 mno	6 pqr
7 stu	8 vwx	9 yz

| 1	2	3
ajs	bot	cpu
4 dkv	5 hmz	6 gl
7 enw	8 fqx	9 iry

- In the left keypad, "hello" can be typed using the following button presses: [3] twice (prints 'h'), [2] twice (prints 'e'), [4] thrice (prints 'l'), [4] thrice (prints 'l'), [5] thrice (prints 'o'). Thus, total number of button presses = 2 + 2 + 3 + 3 + 3 = 13.
- In the right keypad, "hello" can be typed using the following button presses: [5] once (prints 'h'), [7] once (prints 'e'), [6] twice (prints 'l'), [6] twice (prints 'l'), [2] twice (prints 'o'). Thus, total number of button presses = 1 + 1 + 2 + 2 + 2 = 8.

The *keypad click count* is defined as the number of button presses required to print a given string. In order to send messages faster, customers tend to set the keypad design in such a way that the *keypad click count* is minimized while maintaining its validity.

Given a string *text* consisting of lowercase English letters only, find the minimum *keypad click count*.

```
Language  C
                                        Autocomplete Ready
1 > #include <assert.h
15
16    /*
17     * Complete the '
          function below.
18     *
19     * The function i
20     * The function a
21    */
22
23    int minimumKeypad
24
25    }
26
27 > int main() ...
```

Function Description

Complete the function *minimumKeypadClickCount* in the editor below.

minimumKeypadClickCount has the following parameters:
  string text: the string to be typed

Returns:
  int: the minimum keypad click count to type the given string

Constraints

- $1 \leq |text| \leq 10^5$
- *text* consists of lowercase English letters only.

▶ **Input Format For Custom Testing**

▼ **Sample Case 0**

Sample Input For Custom Testing

STDIN	FUNCTION
-----	--------

```
1 > #include
15
16   /*
17    * Compl
        function
18    *
19    * The f
20    * The f
21   */
22
23   int mini
24
25   }
26
27 > int main
```

Test Results

Type here to search

Q) 6th August 2022

2. Code Question 2

1h 6m left

As an intern at Amazon, you have been assigned a task to implement the sign-in pages in the Amazon Dummy Website.

There are three sign-in pages, each with its own API:

	Register	Login	Logout
Function	Registers a new user with the username and password	Verifies the username and password, then grants or denies access	*username* logs out of the website
API Request	register \<username\> \<password\>	login \<username\> \<password\>	logout \<username\>
Returns	• If the registration was successful, Registered Successfully • If the user already exists	• If the login was successful, Logged In Successfully • If the login was unsuccessful, Login Unsuccessful	• If the logout was successful, Logged Out Successfully • If the given username wasn't logged in, Logout

Language **C**

✓ Autocomplete Ready

```
19   |
20   /*
21    * Complete the 'impleme
22    *
23    * The function is expec
       STRING_ARRAY.
24    * The function accepts
       parameter.
25   */
26
27   /*
28    * To return the string
       you should:
29    *      - Store the size
       returned in the result_
30    *      - Allocate the a
       dynamically
31    *
32    * For example,
33    * char**
       return_string_array_us
       result_count) {
34    *      *result_count
35    *
```

Test Results   **Custom**

API Request	login <username> <password>	logout <username>	Language **C**

Returns	• If the registration was successful, Registered Successfully • If the user already exists, Username already exists	• If the login was successful, Logged In Successfully • If the login was unsuccessful, Login Unsuccessful	• If the logout was successful, Logged Out Successfully • If the given username wasn't logged in, Logout Unsuccessful

```
19
20   /*
21    * Complete the 'im
22    *
23    * The function is
     STRING_ARRAY.
24    * The function acc
     parameter.
25    */
26
27   /*
28    * To return the st
     you should:
29    *    - Store the
     returned in the res
30    *    - Allocate
     dynamically
31    *
32    * For example,
33    * char**
     return_string_arra
     result_count) {
34    *    *result_cou
35    *
```

Given a log of API requests, return the list of returns from the mock website.

Notes:

• Initially, there are no users registered.
• If a user is already logged in and makes a login request, the new request is unsuccessful. The original login remains active.
• Each log is an API request and is in one of the three allowed formats.
• The order of execution of each request is the same as the order of input.

Example
The website receives the following API requests in order:

• register user05 qwerty
  ○ A new user with the username "user05" is registered to the website with the password "qwerty".
  ○ The return value is Registered Successfully
• login user05 qwerty
  ○ There is a user with that username.
  ○ The password matches that user's password.
  ○ The user is not already logged in.
  ○ The login is successful and the return value is Logged In Successfully
• logout user05
  ○ The user with username "user05" is currently logged in.
  ○ The logout is successful and the return value is Logged Out Successfully

Return the array ["Registered Successfully", "Logged In Successfully", "Logged Out Successfully"].

Function Description

```
19   |
20   /*
21    * Complete
22    *
23    * The funct
     STRING_ARRAY
24    * The funct
     parameter.
25    */
26
27   /*
28    * To return
     you should:
29    *    - Sto
     returned in
30    *    - All
     dynamically
31    *
32    * For examp
33    * char**
     return_strin
     result_count
34    *    *resu
```

Q) 4th August 2022

Given a list of cities in a 2D universe, find the number of worlds in the universe.

Any pair of cities in the same world have distance <= 10000 between them

All cities in different worlds have distance > 10000

eg:

cities -> ((0,0), (2,3), (4,2), (20000,1), (20000,3), (20002,5))

Ans: Number of worlds = 2

Q) 2nd August 2022

1. You are shopping on Amazon.com for some bags of rice. Each listing displays the number of grains of rice that bag contains. You want to buy a perfect set of rice bags from entire search results list riceBags. A perfect is defined as:
- The set contains at least two bags of rice.
- when rice bags in set perfect are sorted in increasing ordery by grain count, it satisfies the condition;
  perfect[i] * perfect[i] = perfect[i+1] for all 1≤ i <n. Here n is the size of the set and perfect[i] is the number of rice grains in bag i.

Find the largest possible set perfect and return an integer, the size of that set. If no such set is possible, then return -1. It is guaranteed that all elements in riceBags are distinct.

Example: Let the bags of rice available on Amazon have grain counts[3,9,4,2,16]. The following are the perfect sets.

- Set perfect = [3,9], the size of this set is 2.
- Set perfect = [4,2], the size of this set is 2.
- Set perfect = [4,16], the size of this set is 2.
- Set perfect = [4,2,16], the size of this set is 3.
  the size of the largest set is 3.

Class Result{

/*

*Complete the 'maxSetSize' function below.

*

*The function is expected to return an INTEGER

*The function accepts INTEGER_ARRAY riceBags as parameter.

*/

```java
public static int maxSetSize(List<Integer> riceBags){
// write your code here

}


}
```

Q) 1st August 2022

Harry and Potter took a word string. Harry chose a number M (less than the length of the string) and Potter chose N (less than the length of the string). Harry will cut M alphabets from the end of the string and then add it to the beginning and will give it to Potter. Then, Potter will also cut N alphabets from the end of the string, add it to the beginning and ther give to Harry. This process will continue till they get the original word string back.

For a given string and given values of M and N, find the number of turns in which they wi get the original word string back.

Input Specification:

input1: Original word string

Value of M

Q) 24th July 2022

0 Subsequences

≡ Coding

DESCRIPTION

Problem Statement

You are given an integer **n.**

You have to return the length of the shortest string which contains string **"zero"** as a subsequence of the string and the occurrence of the subsequence is **greater or equal to n.**

For example, if n = 2, then possible strings with minimum length are, **"zzero", "zeero", "zerro"** and **"zeroo"**, so here you have to print 5.

Input Format

- The first line of input contains the integer **n.**

Constraints

- $1 <= n <= 10^9$

Output Format

- Return the minimum length of the string that at least n subsequences "zero".

Evaluation Parameters

- Sample Input

- $1 <= n <= 10^9$

Output Format

- Return the minimum length of the string that at least n subsequences "zero".

Evaluation Parameters

- Sample Input

```
2
```

- Sample Output

```
5
```

- Explanation

Given in the problem statement.

EXECUTION TIME LIMIT

2 seconds

Q) 18th July 2022

Alice and difference

Alice is given an array *arr* of length *N*. Alice wants to pick a subset *S* of size *3* from the array *arr* such that the value of

$3 * abs(median\ of\ S - mean\ of\ S)$ is minimized.

Task

Print the minimum value of the above expression that *Alice* can get.

Example

Assumptions

- *N* = 5

- *arr* = [1, 4, 5, 8, 9]

Approach

- You can choose *S* = [1, 5, 9]. In this case, the median of *S* is 5 and the mean of *S* is 5.

- So the value $3 * abs(median\ of\ S - mean\ of\ S)$ is 0.

- Thus, the answer is 0.

$1 \le T \le 10$

$3 \le N \le 10^3$

$0 \le arr[i] \le 10^9$

Code snippets (also called starter code/boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input

```
1
4
2 5 6 11
```

Sample output

```
1
```

Q) 18th July 2022

Substring score

You are given a string S of length N, which contains only lowercase alphabets from 'a' to 'z'.

Let's define the score of a string as xor of the occurrence of each character in the string.

You are required to handle two types of queries:

- 1 L R: You need to answer the score of the *substring* of S from L to R

- 2 X Y: Update the character at position X to Yth *alphabet from* 'a' to 'z'

Notes

- 1-based indexing is followed.
- The XOR operation takes two numbers as operands and does XOR on every bit of two numbers. The result of XOR is 1 if the two bits...

Notes

- 1-based indexing is followed.

- The *XOR* operation takes two numbers as operands and does *XOR* on every bit of two numbers. The result of *XOR* is 1 if the two bits are different, otherwise, it is 0.

  - Y^{th} alphabet from *'a'* to *'z'* is the character at Y^{th} index (1-based) in string "abcdefghijklmnopqrstuvwxyz". For example, 4th alphabet is *'d'*.

  - A substring of a string is a contiguous subsequence of that string.

Task

For each query of type 1, output the score of the string.

Example

Assumptions

- S = "abda"
- Q = 2

Task

For each quer▮▮▮ type 1, output the score of the string.

Example

Assumptions

- S = "abda"
- Q = 2
- queries = [[2,3,2],[1,1,3]]

Approach

- The first query:
    - Update *3rd* index from left to *2nd* alphabet from 'a' to 'z'. So, we change 'a' to 'b'. The resulting string is "abba"
- The second query:
    - The substring from *1* to *3* is *abb*. The frequency of character 'a' is 1 and the frequency of character 'b' is 2
    - Thus, the score of the substring is $1 \oplus 2 = 3$

Function description

Constraint

$1 \leq T| \leq 10$

$2 \leq |S| \leq 10^5$

$1 \leq Q \leq 10^5$

$1 \leq L \leq R \leq |S|$

$1 \leq X \leq |S|$

$1 \leq Y \leq 26$

Code snippets (also called starter code/boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input

```
1
tti
4
2 3 1
1 1 2
2 1 2
1 2 3
```

Sample output

```
2 0
```

Q) 9th July 2022

Minimum number of swaps to put smallest element first & largest last.

Q) 9th July 2022

Q) 2nd July 2022

Q) 2nd July 2022

Q) 1st July 2022

Q) 20th June 2022

Good string

You are given a string S of length N, Q ranges of the form $[L, R]$ in a 2D array *range*, and a permutation *arr* containing numbers from 1 to N.

Task

In one operation, you remove the first unremoved character as per the permutation. However, the positions of other characters will not change. Determine the minimum number of operations for the remaining string to be *good*.
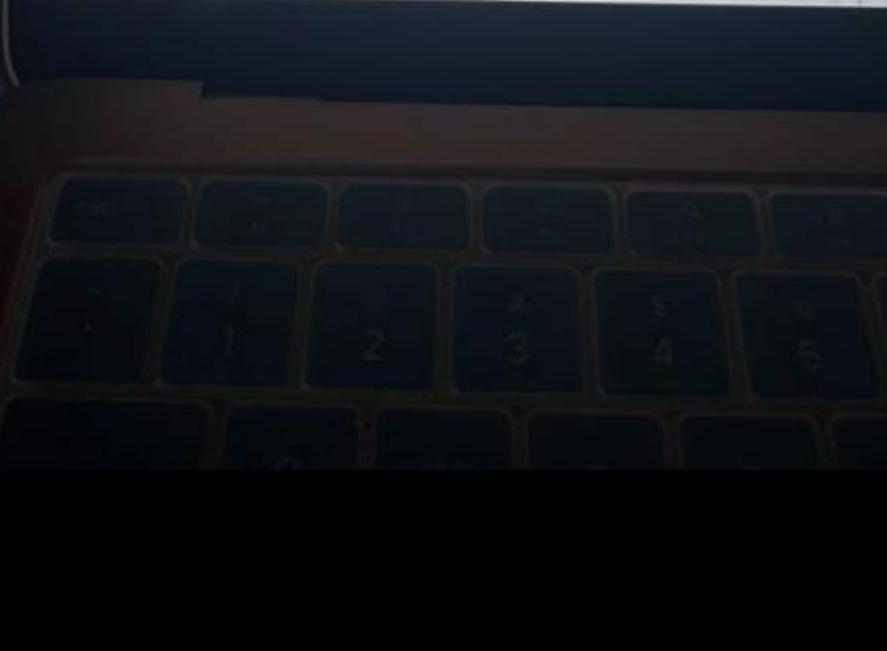
Notes

- A string is considered *good* if all the Q ranges have all distinct characters. Removed characters are not counted.
- A range with *all characters removed* is considered to have all distinct characters
- The sequence of n integers is called a permutation if it contains all integers from 1 to n exactly once
- 1-based indexing is followed

Example

Assumptions

- $N = 5$, $Q = 2$, $S = $ "aaaaa"
- *arr* = [2, 4, 1, 3, 5]
- *ranges* = [[1,2],[4,5]]

0 / 2 Completed ━━━━━━━━━

Example

Assumptions

- $N = 5, Q = 2, S = $ "aaaaa"
- $arr = [2, 4, 1, 3, 5]$
- $ranges = [[1,2],[4,5]]$

Approach
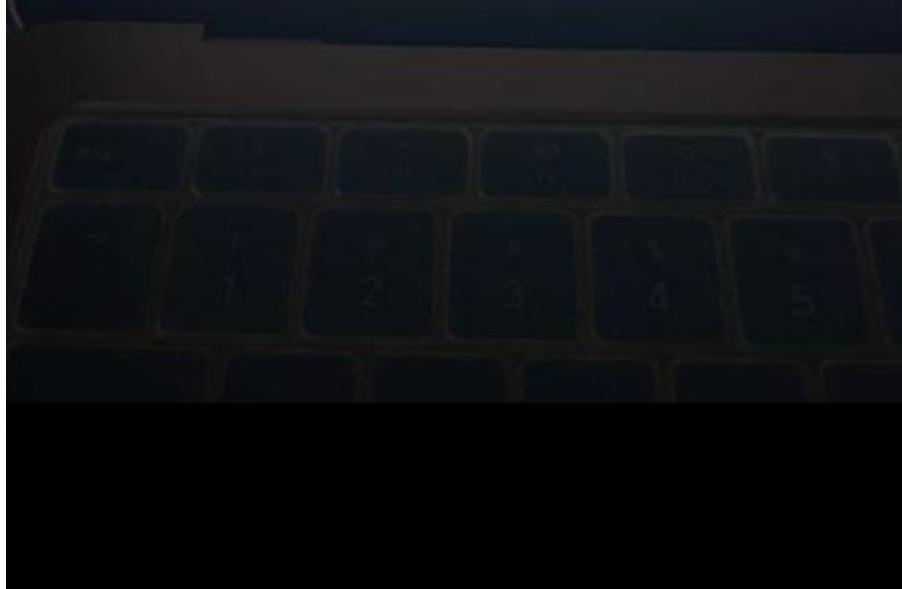
- After the first operation, the string becomes a_aaa
- After the second operation, the string becomes a_a_a
- Now, in both ranges, all characters are distinct.

Hence, the output is *2*.

Function description

Complete the *goodString* function provided in the editor. This function takes the following 6 parameters and returns the minimum number of operations:

- *N*: Represents the length of the string
- *S*: Represents the string
- *arr*: Represents the permutation according to which characters will be removed
- *Q*: Represents the number of ranges
- *ranges*: Represents an array of 2 integer arrays describing the

amazon   0 / 2 Completed ━━━━━━━━

- *Q.* Represents the number of ranges.
- *ranges.* Represents an array of 2 integer arrays describing the ranges *[L, R]* which should have all distinct characters.

Input format

Note: This is the input format that you must use to provide custom input (available above the Compile and Test button).

- The first line contains a single integer *T* denoting the number of test cases. *T* also specifies the number of times you have to run the *goodString* function on a different set of inputs.
- For each test case:
    - The first line contains 2 space-separated integers *N* and *Q*.
    - The second line contains the string *S*.
    - The third line contains *N* space-separated integers denoting the permutation *arr.*
    - Each of the *Q* following lines contains 2 space-separated integers describing the range, *L* and *R*

Output format

For each test case, print a single integer in a single line denoting the minimum number of operations required for the remaining string to be *good.*

Constraints

$$1 \leq T \leq 10$$

0 / 2 Completed

02:59:

minimum number of operations required for the remaining string to be good.

Constraints

$$1 \leq T \leq 10$$

$$1 \leq N \leq 10^5$$

$$1 \leq Q \leq 10^5$$

$$1 \leq arr_i \leq N$$

$$1 \leq L_i \leq R_i \leq N$$

The string S only has lowercase English alphabets.

Code snippets (also called starter code/boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input

```
1
8 3
abbabaab
6 3 5 1 4 2 7 8
1 3
4 7
3 5
```

Sample output

```
5
```
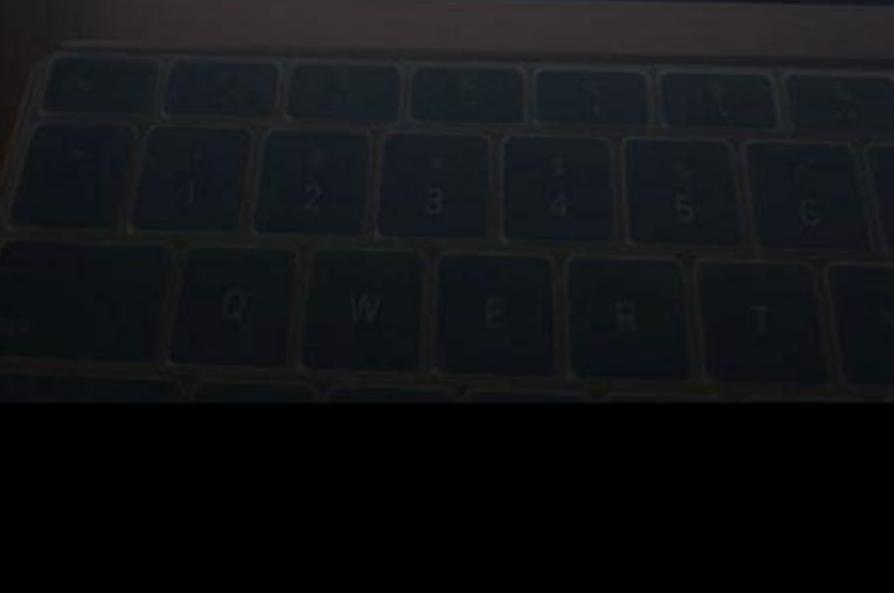
New Submissio

Auto-cor

```
1   #in
2   #in
3   #in
4
5   int
6
7
8   }
9
10  int
11
12
13
14
15
16
17
18
19
20
21
22
```

Tes
C

Q) 14th June 2022

```
Minimum swaps to sort an array
[2, 4, 3, 1, 6] -- 3 swaps
[3, 2, 1] -- 3 swaps
[4, 7] -- 0 swap
[7, 4] -- 1 swap
```

Q) 14th June 2022

Amazon warehouse has a group of n items of various weights lined up in a row. A segment of contiguously placed items can be shipped ogether if only if the difference betweeen the weihts of the heaviest and lightest item differs by at most k to avoid load imbalance.

Given the weights of the n items and an integer k, fine the number of segments of items that can be shipped together.

Note: A segment (l,r) is a subarray starting at index l and ending at index r where l less than equal(<=) r.

Example:
weights = [1, 3, 6], k=3

weight difference between max and min for each (l,r) index pair are:

(0,0) -> max(weights[0]) - min(weights[0]) = max(1)-min(1) = 1-1 =0
(0,1) - > max(weights[0],weights[1]) - min(weights[0],weights[1])= max(1,3)-min(1,3)=3-1=2
(0,2) - > max(weights[0],weights[1],weights[2]) - min(weights[0],weights[1],weights[2])= max(1,3,6)-min(1,3,6)=6-1=5
(1,1) -> max(weights[1]) - min(weights[1]) = max(3)-min(3) = 3-3 =0
(1,2) -> max(weights[1],weights[2]) - min(weights[1],weights[2]) = max(3,6)-min(3,6) = 6-3 =3
(2,2) -> max(weights[2])-min(weights[2]) = max(6)-min(6) = 6-6 =0

as only 5 out 6 pair, is less than equal equal to k (3) , so the number of segments that can shipped together is 5.

Constraints
-- 1<=k, weights[i] <=10^9
-- 1 <= n <=3*10^5

Q) 13th June 2022

https://leetcode.com/discuss/interview-question/1982251/Amazon-OA-or-USA-or-SDE2-or-Minimum-Swaps-To-Make-A-Binary-String-Palindrome

https://leetcode.com/problems/total-appeal-of-a-string/

Q) 10th June 2022

Minimum money

You want to perform N transactions. Every transaction has a cost and a return value.

Suppose you have M amount of money currently. Then, a transaction with cost C and return value R appears. To perform this transaction, clearly, you should have $M \geq C$. If you have less amount than C, then you cannot perform any further transactions. If you have sufficient money, then you *must* perform this transaction and you are left with $M - C + R$ amount of money and move to the next transaction (if there is any). The information about transactions is given in an array *arr* of size $N \ast 2$. The first element is the cost and the second element is the return value of the transactions.

However, you *do not know* the *order* in which N transactions would appear.

Task

Determine the *least* amount of money you should start with so that

next transaction (if there is any). The information about transactions is given in an array *arr* of size *N*2*. The first element is the cost and the second element is the return value of the transactions.

However, you *do not know* the *order* in which *N* transactions would appear.

Task

Determine the *least* amount of money you should start with so that you can always perform all the *N* transactions *irrespective* of the order in which they would appear.

Note: 1-based indexing is followed.

Example

Assumptions

- *T = 1*
- *N = 3*
- *arr = [[3, 1], [4, 2], [6, 0]]*

Approach

- $T = 1$
- $N = 3$
- $arr = [[3, 1], [4, 2], [6, 0]]$

Approach

- Suppose the transaction appears in the order as they are given. Suppose, you start with the amount of *10:*
    - After first transaction we are left with *10 - 3 + 1 = 8*
    - After second transaction we are left with *8 - 4 + 2 = 6*
    - After third transaction we are left with *6 - 6 + 0 = 0*
    - Hence, you are able to perform all *N* transactions given in order *[3, 1], [4, 2], [6, 0].*
- However, if the transaction appears as follows: *[6, 0], [3, 1], [4, 2].* You can check that you would need a minimum amount of *12* in this case. If you start with an amount of *11,* you won't be able to perform the last transaction.
- Therefore, the minimum amount required is *12.*

Function description

$$1 \leq N \leq 2 \times 10^5$$

$$0 \leq arr[i][1], arr[i][2] \leq 10^9$$

Code snippets (also called starter code/boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input ⤷   🗐   **Sample output**   🗐

```
2
3
3 1
4 2
6 0
2
0 5
5 0
```

```
12
5
```

Explanation

The first line contains the number of test cases, $T = 2$.

- If the first transaction appears first, we will get an amount of 5 and then we can perform the second transaction.

- Suppose the second transaction comes earlier. We need at least an amount equal to 5 to complete it.

- Therefore, the minimum amount required is 5.

> ⓘ The following test cases are the actual test cases of this question that may be used to evaluate your submission.

Sample input 1 ⤷   🗐   **Sample output 1**   🗐

```
10
6
7 9
7 6
6 2
6 9
7 10
7 9
8
7 9
```

```
12
13
15
10
13
10
21
12
21
9
```

View more

Q) 10th June 2022

given an array return the count of longest continuous numbers.

Exp: arr={1,2,93,94,3,1001,1000}. o/p: 3

Q) 10th June 2022

Zig-zag traversal of Binary Tree

Q) 6th June 2022

https://leetcode.com/discuss/interview-question/2034486/Prefix-Sum-Amazon-OA

Q) 3rd June 2022

https://leetcode.com/discuss/interview-question/2068122/Amazonor-OA

Q) 3rd June 2022

https://leetcode.com/discuss/interview-question/1988635/Amazon-or-Phone-Screen-or-Array-Strictly-Increasing-Order

Q) 25th May 2022

Find the minimum operations to be performed on the array to have maximum element in the sliding window of 3 to be greater than K. The only allowed operation would be to increase the element by 1.

Example :

Input : array = [1, 3, 0, 3, 1] , K=5

Output: 4

Explanation : Increasing the element at index (0-based Index) 1 and 3 two times.

Q) 25th May 2022

Find the minimum number that can be XOR-ed to each element in the given sorted array to arrange the array to be sorted in descending order

Input : [2, 2, 4, 5]

Output : 5

Explanation : If each element is XOR-ed with 5, will yield the following array : [7, 7, 3, 0]

Q) 24th May 2022

Given a binary string write an algorithm to calculate minimum number of swaps required to make it a palindrome for eg 11101 requires on swap between 3rd and 4th to make it 11011

Q) 24th May 2022

Given a password determine the strength of the password which is calculated by getting substrings in password and calculating strength based on number of unique characters in the substring and adding all the strength

Eg

Good

g - 1

o - 1

o - 1

d -1

go - 2

oo - 1

od- 2

goo - 2

ood - 2

good - 3

total = 16

Q) 23rd May 2022

Determine the min distance required for the robot to remove the obstacle

Input is given as a 2D array which consists of 0, 1 and 9

9 is the obstacle, can pass through 1 and cannot pass through 0

Robot can move top, left, right and bottom

Input: [[1,0,0],[1,0,0],[1,9,1]]

output: 3

Q) 22nd May 2022

https://leetcode.com/discuss/interview-question/1998840/Amazonor-OA-or-Minimum-Days-to-Deliver-All-Parcels

Q) 22nd May 2022

https://leetcode.com/discuss/interview-question/1636493/Amazon-or-OA-or-Max-Length-of-Valid-Server-Cluster

Q) 10th May 2022

Given an array consisting of N integer and two number k,d.

Task1-From the given array we can choose k consecutive elements one after in array and after that leave an array element.Then we calculate maximum sum of all such consecutive element containing subarray.

Task2-From the given array we can choose k+d consecutive elements one after in array and after that leave an array element.Then we calculate maximum sum of all such consecutive element containing subarray.

Find the difference between task1 and task 2 ans.

ex 1->arr=[1,2,3,4,8,9,10]

k=2,d=1;

Task1 elements {3,4}+{9,10}=7+19=26

Task2 elements {1,2,3}+{8,9,10}=6+27=33;

output=33-26=7

Q) 4th May 2022

You are given a tree with V vertices numbered from 1 to N.
The ith edge connects Vertex xi and Vertex yi bidirectionally. You
have to divide this tree into three connected components by cutting
any two edges of the tree. Let the three components be C1,
C2 and C3. Let X1, X2 and X3 be the XOR of all the vertices of the
components C1, C2 and C3 respectively.
Task
Minimize the difference between the maximum and minimum xor
values of the components. In short, you have to minimize the value
of max(X1, X2 ,X3)-min(X1, x2,X3)

Q) 3rd May 2022
https://leetcode.com/problems/3sum/

Q) 3rd May 2022
https://leetcode.com/problems/course-schedule/

Q) 27th April 2022
https://leetcode.com/discuss/interview-question/1655441/amazon-oa

Q) 27th April 2022

https://leetcode.com/discuss/interview-question/1904966/Amazon-orOAorset-7

Q) 16th April 2022

https://leetcode.com/discuss/interview-question/1858858/amazon-oa

Q) 16th April 2022

https://www.chegg.com/homework-help/questions-and-answers/order-ensure-maximum-security-developers-xyz-employ-multiple-encryption-methods-keep-user--q95120166

Q) 7th April 2022

https://leetcode.com/problems/remove-nth-node-from-end-of-list/

Q) 7th April 2022

https://leetcode.com/problems/design-parking-system/

Q) 7th April 2022

https://leetcode.com/problems/missing-number/

Q) 7th April 2022

https://leetcode.com/problems/course-schedule/

Q) 4th April 2022

Q) 2nd April 2022

1. Load Balanced Scheduling

Amazon Web Services (AWS) offers on-demand cloud computing services to its customers.
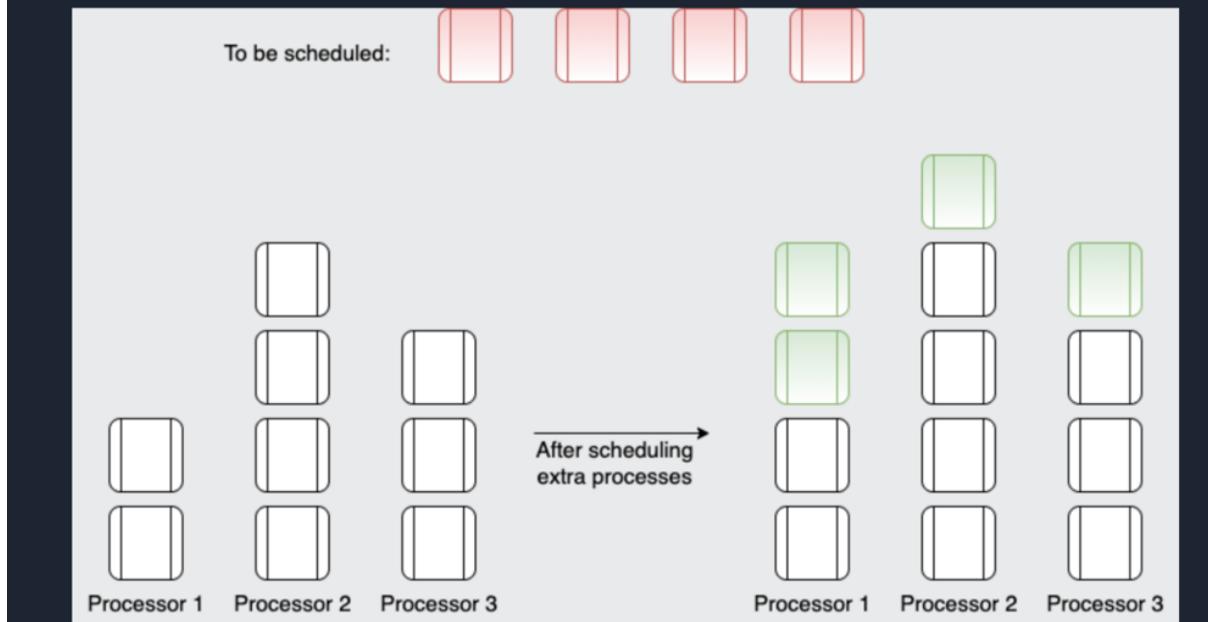
In one of the AWS centers, there are n CPUs, where the ith CPU has $job[i]$ a number of jobs scheduled on it for execution. There are k more jobs that need to be scheduled on these CPUs. For efficient load balancing, these jobs should be added to the CPUs in such a way that the difference between the minimum and a maximum number of jobs scheduled on some CPUs is minimized.

More formally, given an array of n integers, in one move we can choose any element of the array and increment it by 1. The task is to minimize the difference between the maximum and minimum element of the array after exactly k such moves and return this minimum difference.

Example
Consider the number of processors to be $n = 3$, the number of processes scheduled on them to be $job = [2, 4, 3]$, and $k = 4$.
We can add the processes optimally as follows:

The difference between the maximum (= 5) and minimum (= 4) is 1. Hence, the answer is 1. It can be shown that answer cannot be less than 1.
Note that there can be multiple optimal schedules like [5, 4, 4], [4, 5, 4], [4, 4, 5] in the above example, all of which give the same answer.

Function Description
Complete the function *getMinimumDifference* in the editor below.

getMinimumDifference has the following parameters:
  int job[n]: an array of integers denoting the number of processes scheduled on each processor
  long_int k: long integer denoting the number of more processes to be scheduled

Returns
  int: the minimum difference between the maximum and minimum number of processes scheduled on some processor

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq job[i] \leq 10^9$
- $1 \leq k \leq 10^{15}$

Q) 17th March 2022

Given an array of integers -a- and a set of queries of the form [l, r, x], your task is to calculate the number of occurrences of the number x in the inclusive subarray a[l…r] (0-based), for each query, Return the sum of the answers for all queries as the result.

Example

for a=[1,2,1,3,1,2,1] and

arr = [[1,3,3],

[0,4,1],

[2,5,2],

[5,6,1]

]

The output should be solution(queries) = 6.

-The answer to the first query is 1 : the number 3 appears 1 time in the subarray [2,1,3];

-The answer to the second query is 3: the number 1 appears 3 times in the subarray [1,2,1,3,1]:

-The answer to the third query is 1: the number 2 appears 1 time in the subarray [1,3,1,2];

-The answer to the fourth query is 1 : the number 1 appears 1 time in the subarray [2,1] .

-So the answer is 1 + 3 + 1 + 1 = 6.

Input/Output

[execution time limit] 3 seconds (java)

[input] array.integer a

An array of integers.

Guaranteed constraints

$1 <= a.length <= 10^4$

[input] array.array.integer queries

An array of queries, where queries [i] contains three numbers : l , r and x

Guaranteed constraints:

$3 <= queries.length <= 10^5$

queries[i].length = 3,

$0 <= queries[i] [0] <= queries [i] [1] < a.length$

$1 <= queries [i] [2] <= 100.$

[output] integer

The sum of the answers to all queries

Int solution (int[] a , int [] [] queries){

// solution here

```
}
```

Q) 5th March 2022

https://leetcode.com/problems/trim-a-binary-search-tree/

Q) 5th March 2022

https://leetcode.com/problems/minimum-height-trees/

Q) 5th March 2022

https://leetcode.com/problems/house-robber/