DIU Take-Off Programming Contest

Spring 2020

[Main Round]

Organized By

Problem Analysis

Platform Support

Toph

Judging Panel

Special Judge

Mahmud Sajjad Abeer

Trainer

DIU ACM

Daffodil International University

Judging Director

Tanima Hossain

10th Semester

Department of CSE

Daffodil International University

Judges

Md. Erfanul Islam

Nazmus Sakib

Riadh Hasan

Shimul Shutradhar

10th Semester	10th Semester	7th Semester	7th Semester

Index

Problem Name	Setter Name	Reviewer
A. Heart of the Demon	Ahmed Abdullah Shourav	Md. Erfanul Islam, Riadh Hasan
B. Buy Two, Get One Free	Shah Habibul Imran	Rahat Islam Srijon
C. Treat? Itadakimasu…😋	Nazmus Sakib	Md. Albin Hossain
D. Vaccination in BDesh	Md. Shameem Alam	Ahmed Abdullah Shourav
E. Bored in Box	Md. Albin Hossain	Md. Erfanul Islam
F. Tanjiro's Training With Kanao	Rahat Islam Srijon	Nazmus Sakib
G. Inosuke and Trailing Zero	Shimul Shutradhar	Md. Shameem Alam
H. Road to MUZAN	Riadh Hasan	Shah Habibul Imran
I. Here is a treat for You! 🍕🍔	Tanima Hossain	Md. Erfanul Islam, Shah Habibul Imran, Mahmud Sajjad Abeer
J. Behold! The Stopper!!!	Md. Erfanul Islam	Tanima Hossain

A. Heart of the Demon

Category: Pre-Stopper (Data structure)

Problem Setter: Ahmed Abdullah Shourav
Reviewer: Md. Erfanul Islam, Riadh Hasan
Special Thanks: Nazmus Sakib
Analysis:

From the problem statement, we know in each step we need to remove the smallest element from the remaining element and stamina will be used is the number of elements on its left.

Or we can say the cost of removing the smallest element is the number of elements on its left which is greater than this. If j is the index of the smallest element and i is the index of all the elements greater than this where i < j then all the pair (i, j) is called an inversion, so the actual problem is to find the number of inversions in the given array.

Formally, If i <j and ar[i] > ar[j] then the pair (i,j) is called an inversion of the ar[].

Total inversion count of an array can be calculated in several approaches,
1. STL Ordered set (PBDS)
2. Binary Indexed Tree (BIT)
3. Segment Tree
4. Merge sort

We will discuss the Merge Sort solution here,
When we merge two sorted arrays, left and right, we must put the element from right into the new, sorted array if it is smaller than the element on the left. In this situation all the elements in the left array are greater than this element on the right side. We can count the number of inversion in this step, which is the count of the remaining element in the left array. Sum of all of these counts is the inversion count of the whole array. The complexity of this solution is the same as the complexity of merge sort **O(nlog(n)).**

And the middle of the heart is the median of the array, as n is always odd the median is arr[n/2].

B. Buy Two, Get One Free

Category: if-else

Problem Setter: Shah Habibul Imran
Reviewer: Rahat Islam Srijon
Analysis:

If you observe carefully, you may notice that the sum of any number of even integers is always even. Also, the sum of an even number of odd integers is always even. Therefore, the sum of two integers will be even only when one of the integers is odd and another is even. So, all you need to check if there is at least one even integer and one odd integer for you to pick.

C. Treat? Itadakimasu...😋

Category: If-Else+Math

Problem Setter: Nazmus Sakib
Reviewer: Md. Albin Hossain
Analysis:

All we need to do is to find out a number n, made of the last two digits of multiplication of given four numbers. Then If we divide n by 25 then we will get which level it belongs to.

Lets say given numbers are a,b,c,d and **m=a*b*c*d**.

It is a fact that If we divide a number by 10 we get the last digit of that number as remainder. And If we divide a number by 100 we get a number consisting of the last two digits of that number as remainder. Thus, **n = m%100** and **floor(n/25)** is our desired answer.

One drawback of this solution is, we won't be able to store the result of multiplication with int (or even long long int) data type if the given numbers are large enough.

As we know from modular arithmetic, **(a*b)%mod** is the same as **((a%mod)*(b%mod))%mod**.

Thus for this problem we can find the remainder of all four numbers after dividing by 100 and then multiply them. We again divide the result of multiplication to get the last two digits. Thus, we can avoid overflow in the multiplication step and then we determine the level of Katana with the process mentioned above.

So the solution would be:

floor((((a%100)*(b%100)*(c%100)*(d%100))%100)/25).

D. Vaccination in BDesh

Category: Loop

Problem Setter: Md. Shameem Alam
Reviewer: Ahmed Abdullah Shourav
Special Thanks: Tanima Hossain, Md.Erfanul Islam

Analysis:

To solve the problem we need knowledge about how Loop works and some basic arithmetic operations. We need to do the following tasks:

Find each day's spiders who will get fully recovered. And can calculate this using mod(%) operation, i.e **(a_i % K)**
Sum all of N day's results we get from doing the task of above.
So, the answer is: **(a_1 % K) + (a_2 % K) + (a_3 % K) +..................+(a_n % K)**

Time Complexity: **O(T * N)**

E. Bored in Box

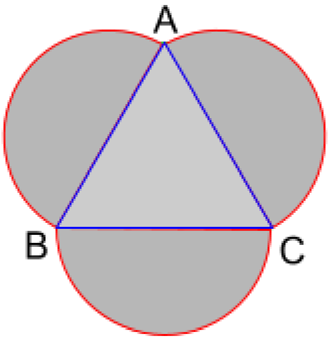
Category: Geometry

Problem Setter: **Md. Albin Hossain**
Reviewer: **Md. Erfanul Islam**
Special Thanks: **Mahmud Sajjad Abeer**
Analysis:

If we subtract the area of the circumcircle from the summation of area of triangle and area of the half circles then we can find the area of dark shaded places.

Haven't got yet?......... Let's see in detail.

Here is our triangle with the half circles on its sides. Let us call it *Figure 1*.

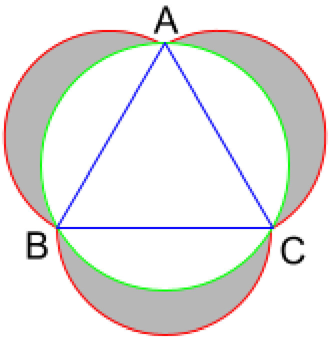


That is, $Figure\ 1\ =\ Triangle\ +\ 3\ \times\ Half\ Circles$

As the triangle is equilateral, the radius of the half circles are also equal.

We have one side of the triangle, AB and we can see that the radius of the *Half Circles* are $\frac{AB}{2}$. Using this information we can find the total area of *Figure 1.*

Now if we draw the circumcircle of the triangle we get our final figure.



So, remaining dark region outside of the circumcircle is our desired answer. We can find it by,

$Dark\ Region = Figure\ 1 - Circumcircle$

N.B: Formulas to calculate the needed areas are given in the problem statement (except one).

F. Tanjiro’s Training With Kanao

Category: Implementation

Problem Setter: Rahat Islam Srijon
Reviewer: Nazmus Sakib
Special Thanks: Md. Erfanul Islam

Analysis:

If read carefully, the problem is very clear and straightforward.
You are given two arrays, you have to tell if there are any **x** and **d** such that **x** is the number of skipped elements and **d** is the interval between elements in the second array, taking the elements, the differences between elements from the first array and second array stays the same all over.
If we do what is told, for all reasonable **x** and **d**, we can easily find if there are such x and d for which the system holds and the constraints are small enough for that to be done.
We can check for **x** and **d** using 3 nested loops.
The first one is for all **x**,
The second one is for all **d**,
The third one is for simulating the situation for that **x** and **d**.
x will be always smaller than **n-m+1**, if not it won’t be valid because at some point **x+i*d** will exceed **n**.
d will be always smaller than **n/(x+(m-1))<n**, if not then **x+i*d** will exceed **n** and become invalid.
And the third loop will iterate over the first array to check differences between respective elements. If all the differences are the same then we can say that we have found such **x** and **d**. On the other hand, if even after checking for all possible **x** and **d**, there was no case where the differences were the same we can be sure that there was no such **x** and **d**.

The overall complexity becomes **O(n*n*m)**.

G. Inosuke and Trailing Zero

Category: Number Theory

Problem Setter: Shimul Shutradhar
Reviewer: Md. Shameem Alam
Analysis:

Trailing zeros are created for multiplication of 10. we need to find out how many 10 are multiplied in the $\frac{resA}{resB}$.

10 prime factorization is. $2^1 \times 5^1$
To make 10, powers of 2 and 5 must be 1. For how many 10 are multiplied in a number we figure out by finding 2 and 5 power in its prime factorization. The minimum power of 2 and 5 will be the how many 10 that are multiplied in the number.
We can express division in the following way.

$$\frac{a}{b} = a^1 \times b^{-1}$$

Now

$$\frac{2^{ae2} \times 5^{ae5}}{2^{be2} \times 5^{be5}}$$

$$\Rightarrow 2^{ae2 - be2} \times 5^{ae5 - be5}$$

Minimum of (ae2 - be2) and (ae5 - be5) will be trailing zero in $\frac{a}{b}$.

Part - 1: In this problem you have to find out power of 2 in resA and resB, After that subtract the power of 2 in resB from the power of 2 in resA.
To calculate the power of 2 in A first, prime factorize each element of A and count the total number of 2 in it that is the power of 2 in resA. Do the same for B to calculate the power of 2 in resB.

Part - 2: After that find power of 5 in resA and resB in the same way. Now, subtract the power of 5 in resB from the power of 2 in resA.

Minimum of subtraction values from part - 1 and part - 2 is the answer.

H. Road to MUZAN

Category: String

Problem Setter: Riadh Hasan
Reviewer: Md. Shah Habibul Imran
Analysis:

In the palindromic string if we split the string in two parts from the middle we have to maintain the same character on both spilled strings that is why we can use any number of even frequency characters in our longest palindromic string.

Now for odd frequency characters we can use only one odd frequency character. For other odd frequency characters we will decrease the frequency by one and add new frequency in our answer string.

Now we can add all even frequency characters and maximum value of odd frequency characters and decrease the value of other odd frequency characters.

time complexity : **O(t*(n+q))**

space complexity: **O(n)**

I. Here is a treat for You! 🍕🍔

Category: Stopper

Problem Setter: Tanima Hossain
Reviewer: Mahmud Sajjad Abeer, Md. Erfanul Islam, Md. Shah Habibul Imran
Analysis:

Observation: Number of techniques for each of them depends on the number of techniques they had in the previous meet. So, it becomes a linear recurrence.
Solution: If you brute force it, it will surely get a TLE as the complexity for that is **O(T×n)** and as there is no pattern in the recurrences we can solve this problem with **matrix exponentiation**. The complexity of the solution becomes **O(T×LogN)**.

J. Behold! The Stopper!!!

Category: Give away

Problem Setter: Md. Erfanul Islam
Reviewer: Tanima Hossain
Analysis:

```
#include <stdio.h>
int main(){
    printf("Never Give Up!\n");
    return 0;
}
```