AMR14C -Editorial

4

PROBLEM LINK:

Practice
Contest

2

DIFFICULTY:

EASY.

PREREQUISITES:

Hashing

PROBLEM:

Given an array, find out pair of numbers such their sum modulo **M** is less than or equal to **X**.

QUICK EXPLANATION:

As the naive solution will have time complexity of **O(n^2)** which will not pass, So we can make a count array A, where A[p] = count of numbers whose value modulo **M** is **p** and we can check if first number module **M** is **p**, then in how many ways we can select another number. The time complexity of above solution will be **O(M+N)**.

EXPLANATION:

Array **A** contains economy rates of the bowlers.

The naive solution will look like:

```
long long int answer = 0;
for(int i =0 ; i < N ; i++)
    for(int j = 0 ; i< N ; j++)
        if( (A[i] + A[j] <= x)
            answer++;
cout<<answer<<endl;
```

But the given solution has time complexity O(n^2) which will not pass in the given time limit.

As you can notice that, the value of **M** is not large and a solution having time complexity of **O(M)** will pass.

An O(M^2) approach:

Make an array **B**. where

B[k] = count of numbers in array **A** which has value of **k** on taking modulo **M**.

Now another naive approach can be written in following manenr having time-complexity of **O(M^2)**.

```
long long int answer = 0;
for(int i = 0 ; i< M ; i++)
    for(int j = 0 ; j< M ; j++)
        if( (i+j)%M <= x)
            answer += B[i]*B[j];
cout<<answer<<endl;
```

In this approach, we are taking all such number, whole modulo **M** value **i** or **j** and if their sum modulo **M** is less than or equal to **x**, then they will contribute to the answer.

The above solution will also not pass, as time complexity of given solution is **O(M^2)**.

Now we will try to optimize the solution to **O(M)**.

as we can observe that if we want **((i+j)%M) <=x**, and we have fix the value of i. then j can take only few contiguous values.

There will be two cases.

$$i + j \leq x$$

$$Case I:$$

$$i \leq x \implies 0 \leq j \leq x - j$$

Case II:

$$i \geq x \implies M - x \leq j \leq M - 1$$

Tags:

easy **×1,220**
hashing **×72**
amr14ros **×33**

Asked: **14 Jan, 23:05**
Seen: **1,344 times**
Last updated: **yesterday**

Related questions

AMR14I - Editorial
AMIFIB - Editorial
Question regarding practice problem Sum Pairs (easy)
[closed] WA [CIELNUM2]
MARCHA1 i'm getting wrong answer ?
Not a Triangle:Wrong answer
easy question:the lead game. So this is a pr straight forward question and I have no i why...
[closed] Enormous input test-pract problem(easy)
Problem Jurassic Park
SGARDEN question showing WA

So, for the case I where i<=x:

```
for(int i=0 ; i<= x; i++)
    answer += B[i]*(B[0] + B[1] + ....+ B[x-i])
```

As we can see that the values which we are multiplying will B[i], their indices are contiguous in nature, so we can use the idea of prefix sum to get the value of the range sum in O(1) time.

Define Pre[i] = B[0] + B[1] + ... + B[i] then B[i] + ... + B[x-i] = Pre[x-i] - Pre[i-1]

So modified code will look like :

```
for(int i=0 ; i<= x; i++)
  answer += B[i]*(Pre[x-i] - Pre[i-1]);
```

Similarly we can handle the second case when **i>=x** . Time complexity of the above solution will be **O(M+N)**, which will easily pass in given time limit.

Editorialist's Solution:

Editorialist's solution can be found here.

amr14ros easy hashing

edited **23 Jan, 22:11**
admin ♦♦
11.4k●346●472●491

asked **14 Jan, 23:05**
amitpandeykgp
249●5●19
accept rate: 0%

@amitpandeykgp Please fix the solution link it's redirecting me to the same page.

Thanks :)

vs13 (15 Jan, 19:30)

Added, thanks.

amitpandeykgp (15 Jan, 21:37)

4 Answers: oldest newest **most voted**

1

Having understood the first and the second approach, I am confused and unable to understand the O(M) approach fully.

```
for (int i = 1; i<= M + X; i++)
          T[i] = T[i-1] + T[i];
      long long ans = 0;
      for (int i = 1; i<=n; i++)
      {
          if (X >= a[i])
              ans = ans + T[X - a[i]];
          ans = ans + (T[M+X-a[i]] - T[M-1-a[i]]);
      }
```

Why is T being calculated up to M+X and when a[i]<=X why is T[M+X-a[i]] - T[M-1-a[i]] also being added to the ans? Could you kindly explain the approach/cases in a little more detail?

Thanks.

Regards, Ankit.

link | award points

answered **16 Jan, 12:22**
ankitdhall
21●2
accept rate: 0%

0

I am not able to understand case 2 of I+j<=x. Please any illustration for that?

link | award points

answered **15 Jan, 12:15**
rudra_sarraf
336●2●9●17
accept rate: 9%

1 In the second case, i>x, So i+j can not be lesser than x. But (i+j)%M <= x which means M<=(i+j) && (i+j)<=M+x
So M-i <=j && j <= M+x-i, as i>x So M-i <=j && j<= M-1 [M-1 is the maximum value of M+x-i (as i>x)], Sorry for typo in latex.

amitpandeykgp (15 Jan, 14:40)

1 Thanks partner. Gotcha :) :)

rudra_sarraf (15 Jan, 19:12)

0

This same code passes here but shows Wrong Answer at here. Can some help.

link | award points

answered **28 Sep, 06:53**
saurabhsuniljain
41●1●4
accept rate: 0%

0

There's bugs in both the recurrences.

1. For i<=x, You should also consider the window (M-i) <= j <= (M-1)
2. For i>x, the j is given as going till M-1. This is wrong. For example if i=x+2, and j = M-1, then (i+j)%M = (x+M+1)%M = (x+1)

which is greater than x.

link | award points

answered **yesterday**

s1d_3
100●2●10
accept rate: 14%

Your answer

[hide preview] community wiki

Type the text

Privacy & Terms

Post Your Answer