CodeChef Discussion

SMPLSUM - Editorial

6

PROBLEM LINK:

Practice
Contest

Author: Kirill Shchetinin
1
Tester: Istvan Nagy
Editorialist: Xellos

DIFFICULTY:

EASY

PREREQUISITES:

Factorisation using the sieve of Erathostenes.

PROBLEM:

You're given many values of N; for each of them, compute $\sum_{K=1}^{N} \frac{N}{gcd(N,K)}$.

QUICK EXPLANATION:

Compute the first few terms by hand, find a simple formula in OEIS. Precompute the smallest prime divisors, use them to factorise N and use that formula to compute the answer.

EXPLANATION:

The most efficient solution to our problem uses Google Search Algorithm. Just compute the first few values by hand and search for this sequence in OEIS. And look, it's here! There are two things to take from the page, which we'll call Result 1 and Result 2:

- the answer is the sum of $d \cdot \varphi(d)$ for all divisors d of N
- the answer is the product of $\frac{p^{2e+1}+1}{p+1} = \frac{(p^e)^2 \cdot p+1}{p+1}$ for all maximum prime powers p^e dividing N

But it sucks to just state those results and provide a link to proofs - let's prove it all here as well!

Result 1 can be seen quite easily: the number of terms with $N/gcd(N,j) = d$ (each term must be a divisor of N, of course) is the Euler totient $\varphi(d)$. That's because we can rewrite this equality to $N/d = k = gcd(N,j) = gcd(kd,j)$, which can only hold if j is a multiple of k - that is, $j = lk$ for $0 < l \le d$. We can divide both sides by k and get $1 = gcd(d,l)$; the number of possible l-s coprime to d and $\le d$ (which is the same as the number of j-s for which $N/d = gcd(N,j)$) is $\varphi(d)$.

In order to prove Result 2, we'll use Result 1. If $N = \prod_j p_j^{e_j}$ (the product goes over prime divisors of N), then any divisor d can be written as $\prod_j p_j^{\alpha_j}$ for any $0 \le \alpha_j \le e_j$. The formula for $\varphi(d)$ is $\prod_j p_j^{\alpha_j - 1}(p_j - 1)$ for $\alpha_j > 0$; it can also be written as $\prod_j \varphi(p_j^{\alpha_j})$, where the factor for $\alpha_j = 0$ is $\varphi(1) = 1$.

Let's take some number $N > 1$ (for $N = 1$, the answer is a product of nothing, which is 1). It'll be divisible by some prime p; let's cut out the max. power of this prime p^e which divides N and write $N = kp^e$. Then, we can take the sum over divisors to be the sum over exponents α of p^α and divisors of k:

$$\sum_{d|N} d\varphi(d) = \sum_{\alpha=0}^{e} \sum_{d|k} dp^\alpha \varphi(dp^\alpha) = \sum_{\alpha=0}^{e} \sum_{d|k} d\varphi(d) p^\alpha \varphi(p^\alpha) = \left(\sum_{\alpha=0}^{e} p^\alpha \varphi(p^\alpha) \right) \left(\sum_{d|k} d\varphi(d) \right) ,$$

where the first sum is

$$\sum_{\alpha=0}^{e} p^\alpha \varphi(p^\alpha) = 1 + \sum_{\alpha=1}^{e} p^\alpha p^{\alpha-1}(p-1) = 1 + \sum_{\alpha=1}^{e} p^{2\alpha} - \sum_{\alpha=1}^{e} p^{2\alpha-1} =$$

$$= \sum_{\alpha=0}^{e} p^{2\alpha} - \sum_{\alpha=0}^{e-1} p^{2\alpha+1} = \frac{p^{2(e+1)}-1}{p^2-1} - p\frac{p^{2e}-1}{p^2-1} = \frac{p^{2e+1}+1}{p+1} = p^{2e} - \frac{p^{2e}-1}{p+1} .$$

The second sum is just the answer for N/p^e (note that p^{2e+1} can overflow, for example if $N = p \approx 10^7$). By continually cutting out primes this way, we eventually arrive at $N = 1$ and obtain the factors of the answer as in Result 2.

It's not very hard to precompute divisors of all numbers up to $N = 10^7$ (there are about $N \log N$ divisors of numbers up to N), then precompute all their totients φ and simulate the sum over divisors, but with our limits, it's too slow. In fact, we can just barely manage to compute one prime divisor of each number using a modified sieve of Erathostenes -

Follow this question
By Email:
You are not subscribed to this question.
subscribe me
(you can adjust your notification settings on your profile)
By RSS:
Answers
Answers and Comments

Question tags:
editorial ×13,984
easy ×3,051
nov15 ×118
factorization ×63
question asked: **14 Nov '15, 19:1**
question was seen: **6,911 times**
last updated: **08 Feb '16, 15:50**

Related questions
CDQU3 - Editorial
MARBLEGF - Editorial
AMR15A - Editorial
ONP - Editorial [Transform Expression]
LETSTOSS - Editorial
CODE1601-Editorial
EZNO - Editorial
Division of problems by difficulty level
CHEFKEY - Editorial
ROBOTG - Editorial

instead of just marking numbers as composite as in the original sieve, we're marking one prime for which they were found to be composite (or primes as their own prime divisors):

```
D[1..N] ={fill with zeroes} # one prime divisor of each number
for i in 2..N:
    if D[i] == 0: # prime
        # mark it as a divisor for all its multiples
        for j in 1..N/i:
            D[i*j] = i
```

This works in $O(N\log N)$ time as well, but with a good constant factor.

With Result 2, however, all we need to do is decompose any N into prime powers. With the precomputed primes, that's easy - as long as $N>1$, we can just keep taking one prime divisor p of N, divide N by it as many times as possible and we've got p^e, with which we'll compute one factor of the answer. Since N decreases quickly when we do this - we always divide it by at least 2, so we'll reach $N=1$ in at most $O(\log N)$ divisions - and computing the factors of our answer as per Result 2 is really fast, this is sufficient to solve the problem.

I'm not sure if I'd be able to make a fast enough solution using only Result 1. Sometimes, optimisation isn't a good idea...

ALTERNATIVE SOLUTION:

Could contain more or less short descriptions of possible other approaches.

AUTHOR'S AND TESTER'S SOLUTIONS:

The author's solution can be found here.
The tester's solution can be found here.
The editorialist's solution can be found here.

RELATED PROBLEMS:

easy editorial nov15 factorization

This question is marked "community wiki".

edited 08 Feb '16, 15:50
admin ♦♦
[18.5k]•348•493•529

asked 14 Nov '15, 19:19
7★ xellos0
[5.9k]•5•41•88
accept rate: 10%

13 Answers:

oldest answers newest answers popular answers

6

it took me 2 days to nail this question , and then 2 days for passing first two test cases , and then 1 day for getting ac in 3rd testcase , and finally i got peaceful sleep on the 5th day...!! :) Nice question (y)

link | award points

answered 16 Nov '15, 15:30
3★ va1ts7_100
[472]•1•6•29
accept rate: 12%

4

Here is a link to my solution https://www.codechef.com/viewsolution/8729843.

I used the property that d*phi(d) is multiplicative. (link https://cs.uwaterloo.ca/journals/JIS/VOL14/Toth/toth9.pdf). So we can just factorise the number and for p^a, the formula can be easily calculated as summation(p^aphi(p^a)) =* summation((p^(2a-1))*(p-1)) which is implemented as simple for loop in my code. So no need of overflow errors.

You can also try LCMSUM and GCDEX on spoj on similar lines.

link | award points

edited 16 Nov '15, 20:31

answered 16 Nov '15, 17:19
6★ likecs
[3.4k]•10•50
accept rate: 9%

@va1ts7_100 ... remove the dot, it's not so hard :\

4★ aragar (16 Nov '15, 20:23)

@aragar done

3★ va1ts7_100 (16 Nov '15, 20:26)

4

My solution : https://www.codechef.com/viewsolution/8767177

I used the fact that the answer for numbers which are powers of a single prime number can be calculated by sum of a gp and for any i, j if they are coprime, ans(i.j) can be written as ans(i).ans(j). Thus the answer for any n can be calculated by precalculating the lowest prime factors of numbers till 10^7.

I was really enlightened by observing that some solutions took as less memory as 2MB, but after the contest ended i observed that they too used an array for precomputation of factors, so how they managed to keep them so memory efficient?

Link to one such solution : https://www.codechef.com/NOV15/status/SMPLSUM,bhardwaj_75

link | award points

answered 17 Nov '15, 11:22
6★ xariniov9
[932]•1•8
accept rate: 8%

1 oops! I observed that, in this solution he freed the memory before returning, so freeing variable memory does not count in the final results?

I always thought they calculate the total memory used on the runtime for entire execution and result displays the maximum memory required.

6★ xariniov9 (17 Nov '15, 11:30)

1

a(p^e) = (p^(2*e+1)+1)/(p+1) there was some overflow in the last test case using this formula , to avoid this overflow we had to use, a(n) = n(n-1)+1,* whenever we encountered a prime n. I had this issue and I am sure many of the people might be getting WA in last case due to overflow. Nice Question !

link | award points

answered 16 Nov '15, 16:27
3★ pra1nay7_313
[11]●1
accept rate: 0%

Yes, I updated that a while ago. You don't have to deal with special cases, just cut out p^{2e} as written above.

7★ xellos0 (16 Nov '15, 16:46)

1

in fact I learn some skills from bhishma's code,he use java,and i use C++. my ac code
Nice question!

link | award points

answered 29 Nov '15, 18:00
2★ ngunaujjj
[11]●1
accept rate: 0%

1

@xellos0

Hey, what is meaning of result 1 "The answer is the sum of d·φ(d) for all divisors d of N".And please can you add some more theory about what approach you are exactly using to solve this problem.

link | award points

answered 09 Dec '15, 14:56
2★ arpit728
[671]●11●47
accept rate: 11%

@xellos0 Reply??

2★ arpit728 (12 Dec '15, 08:31)

1 That equivalently means that the result for a number 'n' is equal to the product of the result of two numbers 'n1' and 'n2' iff gcd(n1, n2)=1 i.e. they are coprime.

If n1 and n2 are further represented as the product of two coprime numbers, you eventually reach at the numbers that are "powers of a single prime".

Now the only task is "how to calculate the result for numbers, which are powers of a single prime?"

This is fairly simple and can be done by summing the phi(d).d for every divisor d of that number.

Have a look on this solution :

https://www.codechef.com/viewsolution/8767177

6★ xariniov9 (13 Dec '15, 20:38)

The basic approach for any problem is to think of the possible solution and observing the patterns while keeping the constraints in the mind at the same time.

When you get some approach, think about it and try to prove the correctness, if you're not able to prove the correctness, try to find a counter example, if you're not able to find one, never hesitate to implement the approach in long challenges. At the end, long challenge will teach you a lot.

6★ xariniov9 (16 Dec '15, 10:45)

@xariniov9 Thanks,I understand how to calculate the answer.I am unable to arrive at the proof can you please simplify the proof.

2★ arpit728 (16 Dec '15, 11:30)

1

Nice editorial !!!

link | award points

answered 16 Dec '15, 21:37
1★ testerprac
[31]●2
accept rate: 0%

0

I think so there would be many users prompting about the failure of last test case even after following the given approach...

I also faced the same issue ... codechef need to answer it efficiently how to clear the Test Case #3 of the question.

Needed the test cases of #3 test case

link | award points

edited 16 Nov '15, 15:18

answered 16 Nov '15, 15:04
3★ jain_mj
[372]●1●9
accept rate: 25%

1 There's a tight time limit, so a constant-efficient algorithm is necessary. However, I needed no optimisations whatsoever this way.

7★ xellos0 (16 Nov '15, 15:09)

I didn't find the time limit very tight, yeah it was a bit tough but the test case 3 was set as such that many users would have got WA in it

3★ jain_mj (16 Nov '15, 15:16)

Now that I remember, there was some overflow; I added a note about it. Is that what you mean?

7★ xellos0 (16 Nov '15, 15:22)

0

I created a array of phi(totients) for all 10^7 numbers and also their lowest prime divisors using these 2 you could solve it. My ACed solution:solution

link | award points

answered 16 Nov '15, 15:25
5★ bhishma
[295]●7
accept rate: 12%

@bhishma - How did you create an array of 10^7 elements? I thought of this same thing, but dropped the idea as C++ doesn't allow creation of int arrays of more than 2*10^6 I think.

5★ s1d_3 (17 Nov '15, 17:06)

I coded in java , and as far as I know we could create arrays of size 10^8 without any issues.

5★ bhishma (17 Nov '15, 18:45)

@s1d_3: Stack limit can kill static arrays. Did you actually try submitting a code with a vector<> of that size?

7★ xellos0 (18 Nov '15, 03:55)

0

Why did calculating primes till 10^7+1 not timed out. I tried this approach but it was taking too long to find the primes even after using the sieve

link | award points

answered 17 Nov '15, 13:59
2★ ab_coding
[1]
accept rate: 0%

Instead of finding primes, you could have found lowest prime factor of each number.

https://discuss.codechef.com/questions/76818/prime-factorization-of-large-numbers

6★ xariniov9 (17 Nov '15, 14:15)

I also applied sieve for 10000000 but i think the complexity is n log(log(n)) and it should work as it is less than 10^8

5★ apptica (17 Nov '15, 16:28)

0

I used different approach. It's obvious that for prime number n value of this function is n * (n - 1) + 1 we can see that for i = 1 to n - 1 gcd (i, n) is 1 and for i = n it's 1. The sum is n * (n - 1) + 1.

This function is multiplicative. for composite number n = p * q where p and q are prime the sum is sum_for_p * sum_for_q i. e. (p * (p - 1) + 1) * (q * (q - 1) + 1).

for composite number that is power of p let's say n = p * p. sum(n) = 1 + p * (p - 1) + p ^ 3 * (p - 1).

for n = p * p * p (i. e. p ^ 3) sum(n) = sum(p * p) + p ^ 5 * (p - 1) and so on.

My Solution

link | award points

answered 18 Nov '15, 00:08
4★ ashok1113
[91]●3
accept rate: 0%

0

Could someone explain the proof of result 1 !

It would help many!!!!

link | award points

answered 19 Dec '15, 19:52
2★ ashishsb95
[1]
accept rate: 0%

0

why can't v compute gcd using recursion and then loop over from i=0....n. it gives same result...plz help..

link | award points

answered 20 Dec '15, 22:28
abhi2811
[1]
accept rate: 0%

Your answer

[hide preview] community wiki:

Preview

reCAPTCHA V1 IS SHUTDOWN
Direct site owners to **g.co/recaptcha/upgrade**

Type the text
Privacy & Terms

Answer the question