riningan's blog

Prime Factorization In log(n) After Sieve

By riningan, 4 years ago,

We use Eratosthenes sieve for prime factorization, storing the primes in an array. But for that, we need to find the primes less than or equal to sqrt(n) which divide n. There are about n/log(n) primes less than or equal to n. So, the complexity is roughly sqrt(n)/log(sqrt(n))*log(n). But if n is asked to be factorized completely where n is within the Sieve range, then we can factorize n is log(n) complexity. And the trick is fairly small. Observe, that, we don't need to run a whole sqrt(n) loop for finding the prime divisors. Instead, we can even store them when n is in the range, say n<= 10^7. But the tricky part is not to store all the prime divisors of n. Let's see the following simulation. Take n = 60. We want to factorize n. We will store the **smallest prime factors only**. This does the trick. If n is composite, then it has such a prime factor, otherwise n is a prime and then the n itself is the smallest prime factor. It is obvious, for any even number n, sp(n)=2. Therefore, we only need to store these primes for odd n only. If we denote the smallest prime factor of n by sp(n), for odd 2 <= n <= 30, we get the following list.

sp(2n)=2, sp(3)=3, sp(5)=5, sp(7)=7, sp(9)=3, sp(11)=11, sp(13)=13, sp(15)=3, sp(17)=17, sp(19)=19, sp(21)=3, sp(23)=23, sp(25)=5, sp(27)=3, sp(29)=29.

Then the factorization is very simple. The optimization is needed only once, when the Sieve() function runs.

```
bool v[MAX];
int len, sp[MAX];

void Sieve(){
        for (int i = 2; i < MAX; i += 2)sp        [i] = 2;//even numbers have
smallest prime factor 2
        for (lli i = 3; i < MAX; i += 2){
                if (!v[i]){
                        sp[i] = i;
                        for (lli j = i; (j*i) < MAX; j += 2){
                                if (!v[j*i])    v[j*i] = true, sp[j*i] = i;
                        }
                }
        }
}

int main(){
        Sieve();
        for (int i = 0; i < 50; i++)    cout << sp[i] << endl;

    return 0;
}
```

Now, notice the difference between the usual prime factorization and this one! The only problem is, you can't use this for n large enough in int range. Still, it seems nice to me and pleasured me when I first found this.

basic math, sieve of eratosthenes

+8 riningan 4 years ago 36

Comments (36)

Write comment?

4 years ago, # | +1

Helgui

[:||||||:] David Gries, Jayadev Misra. A Linear Sieve Algorithm for Finding Prime Numbers, 1978. read this (in Russian)

→ Reply

4 years ago, # ^ | 0

riningan

hmm.

→ Reply

4 years ago, # | 0

dalex

So many minuses, why? It's very useful trick and I don't think that everyone knows it.

→ Reply

3 years ago, # | +5

halfo

Really nice trick! Thanks for sharing.

→ Reply

3 years ago, # | ← Rev. 2 0

savinov

It's better to precalculate not only smallest prime number, but also quotient $cp[i] = i / lp[i]$, to do not unnecessary and TOO SLOW operations of division, especially in case of big number of queries.

→ Reply

3 years ago, # ^ | 0

kien_coi_1997

I think that it is not important. Original source is easy to read and easy to understand. Also, you have to perform divide operations log(n) times only. It seems not too big.

→ Reply

3 years ago, # | 0

mahfuzmohammad

Dude, your tricks is really cool but I think there is some problem in your sample code. **Your Sieve() function doesn't store the smallest prime factors properly**. For 45, the smallest prime factor should be 3 where according to your sample code it stores 5!

→ Reply

3 years ago, # ^ | 0

riningan

that's because I forgot to check first if a number already has a smallest prime divisor. Now it is correct. Thanks for pointing the mistake out.

→ Reply

14 months ago, # | 0

how can we find factorization from sp[]. please explain?

kefaa → Codeforces Round #384 (Div.2)

cgy4ever → Topcoder SRM 702

gKseni → Code Festival 2016: interview and photos from participants

xyz111 → Codeforces Round #254 Editorial

Fighter.human → Graph learning problems

bluemmb → Forgetful Waiter (LA 6354)

kudokidsphr → I want negative feedback, unlike me for god !!!!!

gKseni → Innopolis Open — Olympiad in Informatics for secondary school students: Registration is open until December 15, 2016

arrogantIdiot → Counting Divisors of a Number in $O(N^{\frac{1}{3}})$ [tutorial]

deep9539 → HELP IN DYNAMIC PROGRAMMING (KNUTH OPTIMISATION).

theGalang → Algorithm on euler circuits

zxqfl → Canada Cup Editorial

senthilshravan → 455B A Lot of Games

Arpa → Codeforces Round #383 editorial

tahsynx → help for lightoj 1137 — Expanding Rods

Balajiganapathi → Codeforces problem recommender

hoco → spoj __nested dolls

Xellos → Codeforces Round #333 — editorial

alex256 → Tutorial of Codeforces Round #379 (Div. 2)

anni11 → A silly doubt

Bloger → Hard DP problem.

Arpa → [GYM] Codeforces Round #383 (Hard)

pseudo_empirical → SPOJ Water tricky cases

amd → Algorithm Gym :: Data structures

wanbo → HackerRank 101Hack 44 is coming

Detailed →

akhileshydv20

How can we find factorization from sp[]..please explain?

→ Reply

-Secta-

14 months ago, # ^ | 0

```
vector <int> factorize(int k) {
        vector <int> ans;
        while(k>1) {
                ans.push_back(sp[k]);
                k/=sp[k];
        }
        return ans;
}
```

→ Reply

akhileshydv20

14 months ago, # ^ | 0

Gotcha...thanks :-)

→ Reply

i_love_emilia_clarke

14 months ago, # | 0

How large can MAX be?

→ Reply

Dushyant

14 months ago, # ^ | 0

10^7

→ Reply

i_love_emilia_clarke

14 months ago, # ^ | 0

Hi Dushyant, If the limit is 10^7 then why this code is not working. I have commented out the rest part which is not concerned....

→ Reply

-Secta-

14 months ago, # ^ | 0

Signed integer overflow — http://ideone.com/FXLHXO :)

→ Reply

quantic

6 months ago, # ^ | 0

what should i do for nos of 10^9 range?

→ Reply

Fekete

6 months ago, # ^ | 0

It can be Pollard's "Ro" algorithm or smth like that.

→ Reply

quantic

6 months ago, # ^ | 0

I got Pollard's "Ro" algorithm.really nice one.thank u @fekete

→ Reply

2 months ago, # ^ | 0

Is it a sarcasm?


Fekete

→ Reply


luismo

14 months ago, # | 0

Any problems to solve with this technique ???

→ Reply


SomeRandomGuy

14 months ago, # ^ | 0

http://codeforces.com/contest/546/problem/D

→ Reply


Dushyant

14 months ago, # ^ | ← Rev. 3 0

Medium Factorization

One more

Simple Sum

→ Reply


Taube

13 months ago, # ^ | 0

http://codeforces.com/problemset/problem/222/C

→ Reply

mshibli786

13 months ago, # | 0

hey smallest prime factor for 567 is 3 but you program is outputing 7...plz correct it

→ Reply

Mocking_Jay

13 months ago, # ^ | 0

Sorry but you are mistaken.It is giving 3 as the output.

→ Reply

mshibli786

13 months ago, # ^ | 0

actually i am converting it in java code may be due to i am getting this...if u can convert this in java then it would be very helpful for me and for othes..plz do it soon

→ Reply

mshibli786

13 months ago, # ^ | ← Rev. 2 -6

Whats Wrong With this logic every time exception was occuring or it is Same as ABove logic but not Working for java

```
static void Sieve() {
    for (int i = 2; i < MAX; i += 2)
       sp[i] = 2;// even numbers have smallest
prime factor 2
    for (int i = 3; i < MAX; i += 2) {
       if (!v[i]) {
         sp[i] = i;
         for (int j = i; (j * i) < MAX; j+=2) {
          if (!v[j * i])
                v[j * i] = true;
```

```
            v[j * i] = true;
            sp[j * i] = i;
          }
        }
      }
    }
```

→ Reply

-Secta-

13 months ago, # ^ | 0

```
if (!v[j*i]) v[j*i] = true, sp[j*i] = i;
```

→ Reply

Mocking_Jay

13 months ago, # ^ | ← Rev. 3 0

He has pointed out the mistake.

→ Reply

additya1998

13 months ago, # | 0

This is really nice! Thanks for sharing.

→ Reply

guddu1996

11 months ago, # | ← Rev. 2 0

I don't know why I m getting segmentation fault for the spf() function...
http://codepad.org/cKUBvEJ2

→ Reply

ayu15

3 months ago, # | 0

I am not able to understand that why is it log(n) ???

→ Reply

Flatfoot

3 months ago, # ^ | +6

Consider the prime factorization n = p1 * p2 * ... * pk, where p1, p2, ... pk are the prime factors. n has at most k = log(n) prime factors.

To understand this think of how you can maximize the number of prime factors. You'll get the most number of prime factors for p1 = p2 = ... = pk = 2. So we have n = 2^k. Solving for k yields k = log(n).

→ Reply

ayu15

3 months ago, # ^ | +3

Amazing... thanks :)

→ Reply

prak_blah

4 days ago, # ^ | -10

what is the overall complexity of the Sieve() function mentioned above

→ Reply

2 months ago, # | +3

I think this can be done without extra space :)

→ Reply

saand_nitd
