

Intro CP 6/11/26

Num Theory

① Prime Test

- prime integer ≥ 2 that has exactly 2 divisors, 1 and itself.

```
for (int i=2; i<n; i++)  
    if (n%i == 0) O(n)  
        return false;  
return true;
```

if $n = a \times b$ and $a, b > 1$,
then $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$

Assume $n = a \times b$ and $a > \sqrt{n}$ and $b > \sqrt{n}$
 $> \sqrt{n} \times b$
 $> \sqrt{n} \times \sqrt{n}$
 $= n$

This is a contradiction $n > n$.

Thus our initial assumption must be incorrect. Thus either $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$.

```
for (int i=2; i*i<=n; i++)  
    if (n%i == 0) O(\sqrt{n})  
        return false;  
return true;
```

$\sim 10^{12}$ okay
long long

Prime Sieve

~~2~~ 3 ~~4~~ 5
~~6~~ 7 ~~8~~ 9
11 ~~12~~ 13 ~~14~~ 15
~~16~~ 17 ~~18~~ 19
~~20~~ 21 ~~22~~ 23 ~~24~~ 25

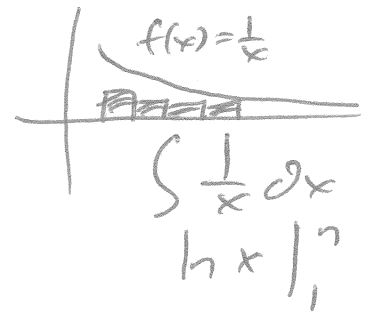
runs about $\frac{n}{i}$ times

$$n \log n \leq \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \dots + \frac{n}{n} \sim n \log n$$

```

vector<bool> isPrime(n+1, true);
for (int i=2; i<=n; i++)
    for (int j=2*i; j<=n; j+=i)
        isPrime[j] = false;

vector<int> primeList;
for (int i=2; i<=n; i++)
    if (isPrime[i])
        primeList.push_back(i);
    
```



Prime Factorization

$$96 = 2^5 \times 3$$

420
 ② 420 → 210 → 105
 +1 +1
 ③ 105 → 35
 +1
 ④ x
 ⑤ 35 → 5
 +1
 ⑥ x
 ⑦ 7 → 7
 +1

```
vector<pair<int,int>> primeFact (ll n) {
```

```
    vector<pair<int,int>> res;
```

```
    ll i = 2;
```

```
    while (i * i <= n) {
```

```
        int exp = 0;
```

```
        while (n % i == 0) {
```

```
            exp ++;
```

```
            n /= i;
```

```
        }
```

```
        if (exp > 0) {
```

```
            res.push_back(pair<int,int> {i, exp});
```

```
        }
```

```
        i ++;
```

```
    }
```

```
    if (n > 1)
```

```
        res.push_back(pair<int,int> {n, 1});
```

```
}
```

```
    return res;
```

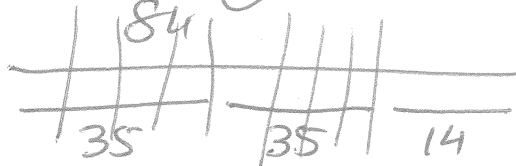
Greatest common divisor Euclidean Algorithm

$$\gcd(35, 84) = 7$$

$$84 = 2 \times 35 + 14$$

$$\gcd(84, 35) = \gcd(35, 14)$$

100 notches



Jump = 24 notches

$$\gcd(100, 24) = 4$$

```
ll gcd(ll a, ll b) {
```

```
    return b == 0 ? a : gcd(b, a % b);
```

```
}
```

Least Common Multiple

$\text{LCM}(a, b)$ is smallest pos int such that both a and b are common divisors of it.

$$a = 14 \quad 14, 28, 42, 56, \textcircled{70}$$

$$b = 10 \quad 10, 20, 30, 40, 50, 60, \textcircled{70}$$

$$\text{LCM}(10, 14) = 70$$

$$\text{gcd}(a, b) * \text{lcm}(a, b) = a * b$$

$$\text{lcm}(a, b) = \frac{a * b}{\text{gcd}(a, b)}$$

$$a = 2^3 * 3^6 * 5^2 * 7^7$$

$$b = 2^1 * 3^{10} * 5^4 * 7^2$$

$$\text{gcd}(a, b) = 2^1 * 3^6 * 5^2 * 7^2$$

$$\text{lcm}(a, b) = 2^3 * 3^{10} * 5^4 * 7^7$$

$$\times \prod_i \frac{\min(a_i, b_i)}{\max(a_i, b_i)} =$$

$$\min(x, y) + \max(x, y) = x + y$$

$$\prod_i \min(a_i, b_i) + \prod_i \max(a_i, b_i)$$

$$\prod_i$$

$$\prod_i a_i * b_i = \prod_i a_i * \prod_i b_i$$

MODULAR INVERSE

$$\text{if } x \cdot y \equiv 1 \pmod{n}$$

then we define $y = x^{-1} \pmod{n}$
"x inverse mod n"

$$\underline{(y \cdot 38)x \equiv (y \cdot 1) \pmod{99}}$$

$$\downarrow$$
$$x \equiv y \pmod{99}$$

$x^{-1} \pmod{n}$ exists if and only if
 $\gcd(n, x) = 1$. Extended Euclidean Alg.
finds $x^{-1} \pmod{n}$ when calculating
 $\gcd(n, x)$.

Divisors of an integer

Options:

#1 brute force checks to $\sqrt{n}$.

```
int res = 0;
```

```
for (int i = 1; i * i <= n; i++) {
```

```
    if (n % i == 0) {
```

```
        res++; // i
```

```
        if (i < n/i) res++; //  $\sqrt{n}$  twice.
```

```
    }
```

```
} return res;
```

What if the int's really big but you have its prime factorization?

$$n = 2^4 \times 3^5 \times 7^1 \times 13^2$$

$$\text{divisor} = 2^a \times 3^b \times 7^c \times 13^d$$

$$0 \leq a \leq 4$$

$$0 \leq b \leq 5$$

$$0 \leq c \leq 1$$

$$0 \leq d \leq 2$$

$$5$$

$$\times 6$$

$$\times 2$$

$$\times 3$$

$$180$$

$$\frac{(a+1)(b+1)(c+1)(d+1)}{(d+1)}$$

Sample Input

10

2 6

3 8

5 1

7 15

11 4

13 100

17 4

19 15

23 5

29 8

find the # of divisors

mod 10^9+7 .

int n;

ll ans = 1;

cin >> n;

for (int i=0; i<n; i++) {

int b,e;

cin >> b >> e;

ans = (ans * (e+1)) % MOD;

}

cout << ans << endl;

Sum of Divisors

$$n = 2^4 \times 3^5 \times 7^1 \times 13^2$$

$$n = 2^3 \times 3^4$$

4x5 divisor

3exp \ 2exp	0	1	2	3	4
0	$2^0 3^0$	$2^0 3^1$	$2^0 3^2$	$2^0 3^3$	$2^0 3^4$
1	$2^1 3^0$	$2^1 3^1$	$2^1 3^2$	$2^1 3^3$	$2^1 3^4$
2					
3					$2^3 3^4$

$$2^0 (3^0 + 3^1 + 3^2 + 3^3 + 3^4)$$

$$\rightarrow 2^1 (3^0 + 3^1 + 3^2 + 3^3 + 3^4)$$

$$\rightarrow 2^2 (3^0 + 3^1 + 3^2 + 3^3 + 3^4)$$

$$\rightarrow 2^3 (3^0 + 3^1 + 3^2 + 3^3 + 3^4)$$

$$= (2^0 + 2^1 + 2^2 + 2^3) (3^0 + 3^1 + 3^2 + 3^3 + 3^4)$$

$$\frac{2^4 - 1}{2 - 1} \times \frac{3^5 - 1}{3 - 1}$$

$$n = 2^4 \times 3^5 \times 7^1 \times 13^2$$

$$\sigma(n) = \left(\frac{2^5 - 1}{2 - 1} \right) \times \left(\frac{3^6 - 1}{3 - 1} \right) \times \left(\frac{7^2 - 1}{7 - 1} \right) \times \left(\frac{13^3 - 1}{13 - 1} \right)$$

sum of divisors

under mod would need mod inverse!

1

$412 = 7 \text{ (2)}$

$2/2 = 1$ (1)

1. 2

⑦ $n = a_1 a_2 a_3$ Prime Factor

$\boxed{2012}$ 2 4 1 $\boxed{15}$ $\boxed{15}$

Indikator 95 + 1 = 1 Indikator 10 302 + 1 Indikator 11

How many 0s are at the end of $n!$?

How times 10 divides into $n!$

$$\begin{array}{c} 2 \times 5 \\ \uparrow \uparrow \\ 4 \end{array}$$

How many times does 5 divide evenly into $n!$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

" n choose k "

$$\begin{aligned} \binom{11}{4} &= \frac{11!}{4!7!} = \frac{\cancel{1} \times \cancel{2} \times \cancel{3} \times \cancel{4} \times \cancel{5} \times \cancel{6} \times \cancel{7} \times 8 \times 9 \times 10 \times 11}{(1 \times 2 \times 3 \times 4) (\cancel{1} \times \cancel{2} \times \cancel{3} \times \cancel{4} \times \cancel{5} \times \cancel{6} \times \cancel{7})} \\ &= \frac{11 \times 10 \times 9 \times 8}{1 \times 2 \times 3 \times 4} \end{aligned}$$

$$\binom{20000}{12345}$$

how many times 2
divides into this