목차

믿음

나누기, 나머지 정리, 약수와 배수, 최대공약수, 최소공배수.. 믿습니다.

합동식

두 정수 a, b와 0이 아닌 정수 n에 대하여 $a \bmod n = b \bmod n$ 다시 말해 $n|(a-b)$라면,

$$a \equiv b \pmod{n}$$

합동식의 성질 1

- $a \equiv a \pmod{n}$
- $a \equiv b \pmod{n}$ 이면 $b \equiv a \pmod{n}$
- $a \equiv b \pmod{n}$ 이고 $b \equiv c \pmod{n}$ 이면 $a \equiv c \pmod{n}$

합동식의 성질 2

$a \equiv b \pmod{n}$ 과 $c \equiv d \pmod{n}$ 를 만족하는 네 정수 a, b, c, d 에 대하여,

- $a \pm c \equiv b \pm d \pmod{n}$
- $ac \equiv bd \pmod{n}$
- 나눗셈은 안돼요.

Examples

- $(a + b) \bmod m = (a \bmod m + b \bmod m) \bmod m$
- $(a - b) \bmod m = (a \bmod m - b \bmod m + m) \bmod m$
- $(a \times b) \bmod m = (a \bmod m \times b \bmod m) \bmod m$
- 나누기는 안돼요.

Naive

- ✓ $i = 2, 3, \ldots, N-1$ 까지 약수가 있는지 확인 : $O(N)$
- ✓ $i = 2, 3, \ldots, \sqrt{N}$ 까지 약수가 있는지 확인 : $O(\sqrt{N})$

Q. 판별하고 싶은 수가 10^5 개면 어떡하나요?

에라토스테네스의 체 (Seive of Era..)

양수 N 이 주어졌을 때, N 이하의 소수를 모두 구할 수 있을까?

- ✓ 2 는 소수, $k \times 2$ 는 모두 소수가 아닙니다.
- ✓ 3 은 소수, $k \times 3$ 은 모두 소수가 아닙니다.
- ✓ 4 는 앞서 소수가 아님을 판단했습니다.
- ✓ 5 는 소수, $k \times 5$ 는 모두 소수가 아닙니다.
- ✓ 6 은 앞서 소수가 아님을 판단했습니다.
- ✓ 7 는 소수, $k \times 7$ 는 모두 소수가 아닙니다.
- ✓ ...

시간 복잡도

- N 이하의 소수 p에 대하여, N 보다 작거나 같은 p의 배수를 제거하는 연산이 필요합니다.
- $\sum_{p \le N} \frac{N}{p} \le N \sum_{k=1}^{N} \frac{1}{k} \approx N \log N$
- 실제로는 $O(N \log \log N)$ 이므로, 거의 선형 시간에 동작합니다.

```
void seive(int N) {
    for (int i=2; i*i<=N; i++) {
        if (not_prime[i]) continue;
        prime.push_back(i);
        for (int j=i*i; j<=N; j+=i) not_prime[j]=1;
    }
}
```

Q. 양수 N 이 가지고 있는 모든 소인수를 출력하시오.

Naive

- $i = 2 \ldots \sqrt{N}$ 까지 돌며 나눌 수 있을 때까지 i 로 나눕니다.
- 시간 복잡도는 $O(\sqrt{N} + \log_2 X)$ 입니다.

```cpp
vector<int> factorize(int N) {
    vector<int> ret;
    for (int i=2; i*i<=N; i++) {
        while (N%i==0) {
            ret.push_back(i);
            N/=i;
        }
    }
    return ret;
}
```

Q. 10^5 개의 양수가 주어졌을 때, 각 양수가 가지고 있는 모든 소인수를 출력하시오.

Smart

- $mp[i] =$ "i를 나눌 수 있는 가장 작은 소수" 라고 정의해 봅시다.
- 이는 에라토스테네스 체를 약간만 수정하면 구할 수 있습니다.

```
void seive_modified(int N) {
    for (int i=2; i<N; i++) mp[i]=i;
    for (int i=2; i*i<=N; i++) {
        if (mp[i]!=i) continue;
        for (int j=i*i; j<=N; j+=i) mp[j]=i;
    }
}
```

Smart

- 현재 가지고 있는 숫자가 x 라면, x 가 포함하고 있는 가장 작은 소인수는 $mp[x]$ 입니다.
- $mp[x]$ 를 정답 배열에 추가하고, $x := \dfrac{x}{mp[x]}$ 를 $x = 1$ 일 때까지 반복합니다.
- 시간 복잡도는 $O(N \log\log N + Q \log_2 X)$ 입니다.

```cpp
vector<int> factorization_fast(int x) {
    vector<int> ret;
    while (x != 1) {
        ret.push_back(mp[x]);
        x = x / mp[x];
    }
    return ret;
}
```

어떤 양수가 가지고 있는 최대 소인수의 개수는 몇 개일까요?

최대공약수 (Greatest Common Divisor, GCD)

- 알 것이라 믿습니다. 공약수 중 최댓값

최소공배수 (Least Common Multiplier, LCM)

- 이 역시 알 것이라 믿습니다. 공통 배수 중 최댓값

GCD의 성질

- $\gcd(a,b) = \gcd(|a|,|b|)$
- $\gcd(a,0) = |a|$
- $\gcd(a,b) = \gcd(b,a)$
- $\gcd(a,b) = \gcd(a+kb,b)$ (단, k는 정수) 는 같이 증명해볼까요?
- $\gcd(a,b) = \gcd(a \mod b, b)$ 도 같이 증명해볼까요?
  - $a \mod b = r$, $a = nb + r$ 인 정수 n이 존재합니다.

LCM의 성질

- 어떤 두 양의 정수 a,b의 최소공배수는

$$\frac{ab}{\gcd(a,b)}$$

유클리드 호제법

- $\gcd(a,b)=\gcd(a \mod b, b)$ 이고, $\gcd(a,b)=\gcd(b,a)$ 입니다.
- WLOG, $b \le a$ 라고 합시다.
- 위 연산을 $\min(a,b)=0$ 일 때까지 반복합니다.
- 시간 복잡도 $O(\log A + \log B)$

```
int gcd(int a, int b) { // a>=b
    if (b==0) return a;
    return gcd(b, a%b);
}
// or else..
// __gcd(a,b), gcd(a,b);
```

~~까먹으세요. 필자의 개인적 취향으로 인한 슬라이드입니다.~~

곱셈적 함수 (Multiplicative function)

서로소 ($\gcd(n, m) = 1$)을 만족하는 두 자연수 n, m에 대하여

$$f(n) \times f(m) = f(nm)$$

을 만족하는 함수를 multiplicative function이라고 합니다.

~~까먹으세요. 필자의 개인적 취향으로 인한 슬라이드입니다.~~

Examples

- 소인수 분해 : $f(n) = \prod_{p|n} p_i^{e_i}$
- 약수의 개수 : $g(n) = \prod e_i + 1$
- 약수의 합 (양의 약수) : $h(n) = \prod_{p|n} 1 + p_i + p_i^2 \ldots p_i^{e_i} = \prod \frac{p_i^{e_i+1} - 1}{p_i - 1}$
- n 과 서로소인 n 이하의 자연수의 개수 (오일러 피 함수) : $\phi(n) = n \prod \frac{p_i - 1}{p_i}$

Q. a^b 를 구하고 싶습니다.

거듭 제곱

- ✓ $b = 0$ 이면 $a^b = 1$
- ✓ $b > 0$ 이면
  - b가 짝수라면, $a^b = a^{\frac{b}{2}} \times a^{\frac{b}{2}}$
  - b가 홀수라면, $a^b = a \times a^{\frac{b-1}{2}} \times a^{\frac{b-1}{2}}$
- ✓ 시간복잡도 $O(\log b)$

나중에 배우게 될 **분할 정복**의 아이디어입니다.

```
ll power(ll a, ll b) {
    if (b==0) return 1ll;
    ll half=power(a,b/2)%MOD;
    return half*half%MOD;
}
```

응용

- 잘 생각해보면.. a의 자료형이 굳이 정수일 필요가 없습니다.
- a가 행렬이라면 두 자료형의 곱셈에 $O(S^3)$이 필요합니다. (S는 행렬 a의 크기)
- 이 경우 시간복잡도는 $O(S^3 \log b)$

알거라 믿습니다.

이항 계수

- $\binom{n}{k} = \frac{n!}{k!(n-k)!}$: 오버플로우의 한계로 인해 최대 20정도 밖에 계산하지 못합니다.
- $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$: 의 점화식을 통해 $O(N^2)$에 구할 수 있습니다.
- 그렇다면 $\binom{100000}{50000}$은 어떻게 구할까요? : 생략하겠습니다.

C++에서 똑똑하게 연산하기

a, b가 정수라면,

- a = (a/b)*b + a%b
- floor와 ceil (b>0)
  - a>=0
    - floor : a/b
    - ceil : (a+b-1)/b
  - a<0
    - floor : (a-b+1)/b
    - ceil : a/b

증명은 골똘히