

#03-3. 에라토스테네스의 체

나정휘

<https://justicehui.github.io/>

목차

소인수분해
곱셈적 함수

소인수분해

에라토스테네스의 체 복습

- 소수 p 의 배수를 지우는 과정을 다시 생각해 보자.
- 만약 x 가 p 에 의해서 처음으로 지워졌으면
- p 는 x 의 가장 작은 소인수
- 가장 작은 소인수가 곱해진 횟수 $e[i]$ 도 알 수 있음
 - $sp[j/i] = i$ 이면 $e[i] = e[j/i] + 1$
 - 그렇지 않으면 $e[i] = 1$

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

```
int SP[1010101], E[1010101];

void Sieve(int n=1'000'000){
    for(int i=2; i<=n; i++){
        if(SP[i]) continue;
        for(int j=i; j<=n; j+=i){
            if(SP[j]) continue;
            SP[j] = i;
            E[j] = SP[j/i] == i ? E[j/i] + 1 : 1;
        }
    }
}
```

소인수분해

소인수분해

- x 의 가장 작은 소인수를 $sp[x]$ 라고 하면
- $\text{while}(x > 1) x /= sp[x];$
- 최대 $\log_2 x$ 번의 나눗셈으로 x 소인수분해 가능
 - x 는 매번 절반 이상 감소

```
#include <bits/stdc++.h>
using namespace std;

int SP[5050505];
void Sieve(int n){
    for(int i=2; i<=n; i++){
        if(SP[i]) continue;
        SP[i] = i;
        for(int j=i+i; j<=n; j+=i) if(!SP[j]) SP[j] = i;
    }
}

int main(){
    ios_base::sync_with_stdio(false); cin.tie(nullptr);
    Sieve(5000000);
    int N; cin >> N;
    for(int i=1; i<=N; i++){
        int X; cin >> X;
        while(X > 1){
            cout << SP[X] << " ";
            X /= SP[X];
        }
        cout << "\n";
    }
}
```

질문?

곱셈적 함수

곱셈적 함수

- 서로소인 두 자연수 n, m 에 대해 $f(n)f(m) = f(nm)$ 이 성립하는 함수 f 를 곱셈적 함수라고 부름
- 소인수분해
 - 소수의 거듭제곱에 대한 함숫값을 빠르게 구할 수 있으면
 - 소인수분해를 이용해 임의의 자연수의 함숫값을 빠르게 구할 수 있음
 - 편의상 n 를 소인수분해해서 $n = \prod p_i^{e_i}$ 로 나타내자.
 - $f(n) = \prod f(p_i^{e_i})$
- 예시
 - 양의 약수의 개수 $\tau(n) = \prod (e_i + 1)$
 - 양의 약수의 합 $\sigma(n) = \prod (1 + p_i + p_i^2 + \cdots + p_i^{e_i}) = \prod \frac{p_i^{e_i+1} - 1}{p_i - 1}$
 - n 이하 자연수 중 n 과 서로소인 개수 $\phi(n) = n \prod (1 - 1/p_i)$
 - 이걸 지금 몰라도 됨

곱셈적 함수

곱셈적 함수의 계산

- 에라토스테네스의 체를 이용하면 $f(1), f(2), \dots, f(n)$ 를 모두 계산할 수 있음

- n 의 최소 소인수를 p_k 라고 하면
$$f(n) = f\left(\frac{n}{p_k^{e_k}}\right) f(p_k^{e_k}) = \frac{f\left(\frac{n}{p_k}\right) f(p_k^{e_k})}{f(p_k^{e_k-1})}$$

```
int SP[1010101], E[1010101], Tau[1010101];
ll Sigma[1010101];

int GetTau(int p, int e){ return e + 1; }
ll GetSigma(int p, int e){ return (Pow(p, e+1) - 1) / (p - 1); }

void Sieve(int n=1'000'000){
    for(int i=2; i<=n; i++){
        if(SP[i]) continue;
        for(int j=i; j<=n; j+=i){
            if(SP[j]) continue;
            SP[j] = i; E[j] = SP[j/i] == i ? E[j/i] + 1 : 1;
        }
    }
    Tau[1] = Sigma[1] = 1;
    for(int i=2; i<=n; i++){
        Tau[i] = Tau[i/SP[i]] / GetTau(SP[i], E[i]-1) * GetTau(SP[i], E[i]);
        Sigma[i] = Sigma[i/Pow(SP[i], E[i])] * GetSigma(SP[i], E[i]);
    }
}
```

질문?