# SỐ HỌC

## 1-Sàng Eratosthen

*Input:* Số nguyên $n$

*Output:*

- E[1..n] : Với E[i] là một ước nguyên tố của i (qui ước E[1]=1)
- nP : Số lượng số nguyên tố trong đoạn [1,n]
- P[...] : Danh sách các số nguyên tố trong đoạn [1,n] với P[i] là số nguyên tố thứ i

```
void Eratosthen(int n) {
    for(int i=1;i<=n;++i) E[i]=i;
    for(int i=2;i<=n/i;++i) if (E[i]==i) {
        for(int j=2;j<=n/i;++j) E[i*j]=i;
    }
    nP=0;
    for(int i=2;i<=n;++i) if (E[i]==i) P[++nP]=i;
}
```

## 2-Phân tích một số thành số nguyên tố

*Input:* int64_t n ($\leq 10^{12}$)

*Output:*

- int k : Số lượng thừa số nguyên tố khác nhau của n
- int64_t x[1..70] : Danh sách các thừa số nguyên tố của n
- int y[1..70] : Lũy thừa tương ứng của x[1..k]

(Chú ý $n = x_1^{y_1} \times x_{2_2}^{y} \times \ldots \times x_k^{y_k}$)

Giả thiết rằng đã có mảng các số nguyên tố P[1], P[2], ..., P[$10^6$] (Sử dụng sàng Eratosthen)

```
void PhanTichSNT(int64_t n) {
    k=0;
    int i=1;
    int64_t a;
    while (n>1000000) {
        while (P[i]<=n/P[i] && n%P[i]!=0) ++i;
        if (P[i]>n/P[i]) a=n; else a=P[i];
        x[++k]=a; y[k]=0;
        while (n%a==0) ++y[k], n/=a;
    }
    while (n>1) {
        a=E[n];
        x[++k]=a; y[k]=0;
        while (n%a==0) ++y[k], n/=a;
    }
}
```

## 3. Kiểm tra một số là số nguyên tố

*Input:* int64_t n ($\leq 10^{12}$)

*Output:* 1 (true) - nếu là số nguyên tố, 0 (false) - không là số nguyên tố

```
int isPrime(int64_t n) {
```

```
    if (n==1) return 0;
    int i=1;
    while (P[i]<=n/P[i] && n%P[i]!=0) ++i;
    if (P[i]>n/P[i]) return 1;
    else return 0;
}
```

### 4. Mảng số lượng ước

*Input:* int n ($\leq 10^6$)

*Output:* int S[maxn] với S[i] là số lượng ước của i ($1 \leq i \leq n$)

```
void SoluongUoc(int n) {
    S[1]=1;
    for(int x=2;x<=n;++x) {
        int p=E[x], y=x, k=0;
        while (y%p==0) ++k, y /= p;
        S[x]=(k+1)*S[y];
    }
}
```

Chú ý công thức: Nếu $n = p_1^{k_1} \times p_2^{k_2} \times \ldots \times p_r^{k_r}$ thì:

$$S(n) = (k_1 + 1) \times (k_2 + 1) \times \ldots \times (k_r + 1)$$

### 5. Mảng tổng các ước

*Input:* int n ($\leq 10^6$)

*Output:* int T[maxn] với T[i] là tổng các ước của i ($1 \leq i \leq n$)

```
void TongCacUoc(int n) {
    T[1]=1;
    for(int x=2;x<=n;++x) {
        int p=E[x], y=x;
        int sum=1;
        while (y%p==0) sum=sum*p+1, y /= p;
        T[x]=sum*T[y];
    }
}
```

Chú ý công thức: Nếu $n = p_1^{k_1} \times p_2^{k_2} \times \ldots \times p_r^{k_r}$ thì:

$$T(n) = \frac{p_1^{k_1+1} - 1}{p_1 - 1} \times \frac{p_2^{k_2+1} - 1}{p_2 - 1} \times \ldots \times \frac{p_r^{k_r+1} - 1}{p_r - 1}$$

### 6. Thuật toán Euclid mở rộng

*Input:* int a, b : a, b $\geq 0$ có ít nhất một số dương

*Output:* int d, x, y : d=gcd(a,b), $d = a \cdot x + b \cdot y$

```
void E_gcd(int a,int b,int &d,int &x,int &y) {
    if (b==0) {
        d=a, x=1, y=0;
        return;
```

```
    }
    int x1, y1;
    E_gcd(b, a%b, d, x1, y1);
    x=y1;
    y=x1-(a/b)*y1;
}
```

## 7. Số học trường đồng dư P ($P \le 10^9$)

a) Phép cộng, trừ:

$$(a \pm b)\ \%\ P = ((a\ \%\ P) \pm (b\ \%\ P) + 2*P)\ \%\ P$$

b) Phép nhân

$$(a * b)\ \%\ P = (int64\_t(a\ \%\ P)*(b\ \%\ P) + int64\_t(P)*P)\ \%\ P$$

c) Phép chia: Tính:

$$\left(\frac{a}{b}\right)\ \%\ P$$

Ở đây giả thiết gcd(b,P)=1 (vì nếu không kết quả sẽ không duy nhất

```
int Chia(int a,int b,int P) {
    int d;
    int x, y;
    E_gcd(b,P,d,x,y);
    return (int64_t(a)*x+int64_t(P)*P) % P;
}
```

d) Phép lũy thừa nhanh: Tính:

$$(a^n)\ \%\ P$$

```
int LuyThua(int a,int n,int P) {
    if (n==0) return 1;
    int t=LuyThua(a,n/2,P);
    t=(int64_t(t)*t) % P;
    if (n%2) t=(int64_t(t)*a) % P;
    return t;
}
```