

Min_25筛 模板

[题目传送门](#)

现在还不是完全明白，意会一下。

```
xxxxxxxxxx

#include<bits/stdc++.h>

using namespace std;

#define il inline

typedef long long ll;

const int N=2e5+5,mod=1e9+7;

bool isp[N]; //判断是否为质数

int cnt,m,id1[N],id2[N];

//cnt质数的个数,m待处理的数个数

//id1[]储存<=sq的m下标, id2[]储存>=sq的m下标

ll n,sq,w[N],p[N],g1[N],g2[N],s1[N],s2[N];

//sq=sqrt(n)

//w[i]第i个待处理的数

//p[i]第i个素数

//g1: f'(k)=k, g2: f'(k)=k^2

//s1 质数前缀和 s2 质数平方前缀和

void Euler() { //欧拉筛

    isp[1]=1;

    for(int i=2;i<=sq;i++) {

        if(!isp[i]) p[++cnt]=i;

        for(int j=1;j<=cnt&&p[j]*i<=sq;j++) {

            isp[i*p[j]]=1;

            if(i%p[j]==0) break;

        }

    }

    for(int i=1;i<=cnt;i++) { //预处理前缀和

        s1[i]=(s1[i-1]+p[i])%mod;

        s2[i]=(s2[i-1]+p[i]*p[i]%mod)%mod;

    }

}

il ll f1(ll x) { //前x项和

    x%=mod;return x*(x+1)/2%mod;
```

```

}

il ll f2(ll x) { //前x项平方和

    x%=mod;return x*(x+1)%mod*(2*x%mod+1)%mod*166666668%mod;

}

il ll ID(ll x) { //获取id

    return x<=sq?id1[x]:id2[n/x];

}

void pre() { //求出所有待处理的数, 计算g1, g2

    for(ll l=1, r; l<=n; l=r+1) {

        r=n/(n/l), w[++m]=(n/l);

        g1[m]=f1(w[m])-1, g2[m]=f2(w[m])-1;

        if(w[m]<=sq) id1[w[m]]=m;

        else id2[n/w[m]]=m;

    }

}

ll S(ll x, int y) {

    if(p[y]>=x) return 0;

    ll ans=(g2[ID(x)]-g1[ID(x)]-(s2[y]-s1[y])+mod*2)%mod;

    for(int i=y+1; i<=cnt&& p[i]*p[i]<=x; i++)

        for(ll e=1, sp=p[i]; sp<=x; e++, sp*=p[i]) //sp=p^e f(sp)=sp*(sp-1)

            ans=(ans+sp%mod*(sp%mod-1)%mod*(S(x/sp, i)+(e>1))%mod)%mod;

    return ans;

}

void DP() {

    for(int i=1; i<=cnt; i++) {

        for(int j=1; j<=m&& p[i]*p[i]<=w[j]; j++) {

            g1[j]=(g1[j]-p[i]*(g1[ID(w[j]/p[i]])-s1[i-1])%mod+mod)%mod;

            g2[j]=(g2[j]-p[i]*p[i]%mod*(g2[ID(w[j]/p[i]])-s2[i-1])%mod+mod)%mod;

        }

    }

}

int main() {

    scanf("%lld", &n), sq=sqrt(n);

    Euler(); pre(); DP();

    printf("%lld\n", (S(n, 0)+1)%mod);

    return 0;

}

```

学习博客 [传送门](#) [传送门](#)

$[1, n]$ 内的素数和 = $g_1[ID(n)]$ 。

[CCPC网络赛例题](#)

```
#include<bits/stdc++.h>

using namespace std;

#define il inline

#define mst(a,b) memset(a,b,sizeof a)

typedef long long ll;

const int N=2e5+5;

bool isp[N];

int cnt,m,id1[N],id2[N],mod;

ll n,sq,w[N],p[N],g1[N],s1[N];

void Euler() {          //欧拉筛

    isp[1]=1;

    for(int i=2;i<=sq;i++) {

        if(!isp[i]) p[++cnt]=i,s1[cnt]=s1[cnt-1]+i;

        for(int j=1;j<=cnt&&p[j]*i<=sq;j++) {

            isp[i*p[j]]=1;

            if(i%p[j]==0) break;

        }

    }

}

il ll f1(ll x) {          //前x项和

    x%=mod;return x*(x+1)/2%mod;

}

il ll ID(ll x) {          //获取id

    return x<=sq?id1[x]:id2[n/x];

}

void pre() {              //求出所有待处理的数,计算g1

    for(ll l=1,r;l<=n;l=r+1) {

        r=n/(n/l),w[++m]=(n/l);

        g1[m]=f1(w[m])-1;

        if(w[m]<=sq) id1[w[m]]=m;

        else id2[n/w[m]]=m;

    }

}
```

```

void DP() {
    for(int i=1;i<=cnt;i++)
        for(int j=1;j<=m&& p[i]*p[i]<=w[j];j++)
            gl[j]=gl[j]-p[i]*(gl[ID(w[j]/p[i])]-s1[i-1]);    //这里不用取模,不然会TLE
}

int main() {
    int t;scanf("%d",&t);
    while(t--){
        m=cnt=0;
        scanf("%lld%d",&n,&mod),n++,sq=sqrt(n);
        Euler();pre();DP();
        printf("%lld\n", (gl[ID(n)]%mod+f1(n)-5+mod)%mod);
    }
    return 0;
}

```