huntian oy (数论&卷积&杜教筛)

思路:

$$f(n,a,b)=\sum_{i=1}^{n}\sum_{j=1}^{i}gcd(i^a-j^a,i^b-j^b)[gcd(i,j)=1]\pmod{10^9+7}$$

$gcd(i^a-j^a,i^b-j^b)$

$i^a-j^a=(i-j)(i^{a-1}+i^{a-2}j+\cdots+j^{a-1})$

$i^b-j^b=(i-j)(i^{b-1}+i^{b-2}j+\cdots+j^{b-1})$

因为a,b互质。

所以$gcd(i^a-j^a,i^b-j^b)=i-j$

$$原式=\sum_{i=1}^{n}\sum_{j=1}^{i}(i-j)[gcd(i,j)=1]$$

$$=\sum_{i=1}^{n}i\times\varphi(i)-\sum_{i=1}^{n}\sum_{j=1}^{i}j[gcd(i,j)=1]$$

$$=\sum_{i=1}^{n}i\times\varphi(i)-(\sum_{i=1}^{n}\frac{i\times\varphi(i)}{2}-\frac{1}{2})$$

$$=\frac{\sum_{i=1}^{n}i\times\varphi(i)-1}{2}$$

然后一股神秘的卷积力量可将式子变为递推式。(这里卷积学的不好待补)

$$S(n)=\sum_{i=1}^{n}i^2-\sum_{i=2}^{n}iS(\frac{n}{i})$$

对于$n\le10^6$的数据可用第一个式子。

否则用递推式记忆化递归。

求$\varphi(i)$ 用欧拉筛即可。

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=5e6+5,M=2e4+5,inf=0x3f3f3f3f,mod=1e9+7;
#define mst(a,b) memset(a,b,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
#define pb push_back
#define il inline
ll inv2,inv6,phi[N];
ll ksm(ll a,ll n){
    ll ans=1;
    while(n){
        if(n&1) ans=ans*a%mod;
        a=a*a%mod;
```

```
        n>>=1;
    }
    return ans;
}
int a[N],p[N],cnt;
void Euler(int n){
     phi[1]=1;
     for(int i=2;i<=n;i++){
          if(!a[i]){
              p[++cnt]=i;
              phi[i]=i-1;
           }
           for(int j=1;j<=cnt&&p[j]*i<=n;j++){
                a[p[j]*i]=1;
                if(i%p[j]==0){
                      phi[p[j]*i]=phi[i]*p[j];
                      break;
                   }
                else phi[p[j]*i]=phi[i]*phi[p[j]];
           }
     }
     for(int i=2;i<=n;i++){
        phi[i]=(phi[i-1]+phi[i]*i%mod)%mod;
     }
}
unordered_map<int,int>mp;
int solve(int n){
    if(n<=N-5) return phi[n];
    if(mp[n]) return mp[n];
    ll tmp=1LL*n*(n+1)%mod*(2*n+1)%mod*inv6%mod;    //这里要用3个mod
    for(int l=2,r;l<=n;l=r+1){
        r=n/(n/l);
        tmp-=1LL*(r-l+1)*(l+r)/2%mod*solve(n/l)%mod;
        tmp=(tmp%mod+mod)%mod;
    }
    return mp[n]=tmp;
}
int main(){
    inv2=ksm(2,mod-2),inv6=ksm(6,mod-2);
    int t;
    Euler(N-5);
    scanf("%d",&t);
    while(t--){
        int n,a,b;
        scanf("%d%d%d",&n,&a,&b);
        int ans=1LL*(solve(n)-1)*inv2%mod;
        ans=(ans%mod+mod)%mod;
        printf("%d\n",ans);
    }
    return 0;
}
```