Solution

facsum

20%

先预处理出来$\varphi(n)$和$\sigma_0(n)$（可以暴力枚举约数来计算），然后再暴力枚举约数算$f(n)$，最后求和。

复杂度$O(n\sqrt{n})$。

50%

法一：先线性筛把$\varphi(n)$和$\sigma_0(n)$筛出来，然后我们可以先枚举d，然后枚举d的倍数，然后算对各个f的贡献。复杂度$O(nlog(n))$。

法二：我们发现这个函数是个积性函数，我们考虑：

$$f(p^k)=p^{km}(k+1-pk)$$

然后，我们在线性筛的过程中，维护一下每个数的最小质因子$minp(i)$，该最小质因子对应的指数$minpk(i)$，以及将该最小质因子p^k那项去掉后剩下的部分$other(i)$。这样我们就可以用积性函数的性质得到：

$$f(p^kc)=f(p^k)f(c)$$

对于$c=1$的情况，我们用上面那个公式暴力算就行了。复杂度上界是：$O(nlog(n))$。

100%

对于每个质数我们先处理出来p^m是多少，然后算$other(i)\neq 1$的情况和上面法二一样，对于$i=p^k$这种项，我们考虑：

$$\frac{f(p^k)}{f(p^{k-1})}=p^m\frac{k+1-pk}{k-pk+p}$$

先线性处理出$[1,n]$所有数的逆元，然后就可以$O(1)$从$f(p^{k-1})$转移到$f(p^k)$。（但有个小case，就是当p=k=2时，分母为0，不能用上面转移，直接暴力算就行了，就只有这一项不满足）。总复杂度$O(n)$。

> TIPS:关于线性求逆：
>
> 法一：先算出$1!,2!,\dots,n!$模意义下的值，然后求一次$n!$的逆元，通过它依次算出$(n-1)!,(n-2)!,\dots,2!,1!$的逆元，然后再算出每个数的逆元。
>
> 法二，考虑：$a\%b=a-(a/b)*b$（这里的$\%,/,*$都是c++中的含义）。然后列成同余式子：
>
> $$a\%b\equiv-(a/b)b \pmod a$$
>
> 同时乘以$(a\%b)^{-1}b^{-1}$，得到：
>
> $$b^{-1}\equiv-(a/b)(a\%b)^{-1} \pmod a$$
>
> 即：$inverse[b]=-inverse[mod\%b]*(mod/b)$。

group

30%

直接暴力算数列，直到遇到一个出现过的数，这样后面出现的数前面一定都出现过了，所以直接看当前出现的数有多少个不同即可。

复杂度：$O(Tmod)$。

100%

因为$gcd(a,mod)=1$，所以本质上那个数列是一个"环"，即是某一段数一直重复。

因为$a^{\varphi(n)}\equiv1$，所以该最小循环节一定是$\varphi(n)$的约数（我们证明过的）。

所以我们可以直接枚举所有$\varphi(n)$的约数，然后暴力check是否有$a^d\equiv1$。

复杂度：$O(T\sqrt{m}\,log(m))$

ccount

20%

把组合数的表打出来，暴力check。

40%

考虑：

$$\binom{n}{m}=\frac{n!}{m!(n-m)!}$$

我们本质就是就是看分子分母中5^k中k是否相等，相等则不是5的倍数，否则就是。

然后我们有：$f(n)=f(n/5)+n/5$，其中$f(n)$表示$n!$中5^k中的k 。

100%

根据lucas定理，可以在5进制下，做数位dp。