彩虹般绚烂 题解

*将 itoaibad 反过来可以得到 da_biao_ti,这提示我们这题需要打表!!!

子任务 1

$$\Pi_{i\le n}\Pi_{j\le\lfloor\frac{n}{i}\rfloor}f(j)^{\lfloor\frac{n}{j}\rfloor}=\Pi_{i\le n}f(i)^{\lfloor\frac{n}{i}\rfloor^2}$$

跑一遍线性筛法，前两个式子可以暴力计算，但后两个式子也暴力算就可能会 T 了，所以预处理出f和g的前缀积再套个分块即可。

子任务 2&3

设$F(i)=i^{i*[i是素数]}$，$G(i)=i^{\varphi(i)}$

结合$k=1$和提示，我们容易想到将$\Pi_{i\le n}F(i)$和$\Pi_{i\le n}G(i)$的结果打表存下。

对于区间$[l,r]$，可以用埃式筛法进行计算，具体一点来讲就是用$\sqrt{r}$内的素数 p 去更新区间$[l,r]$中 p 的倍数，这样就可以得到区间$[l,r]$的素数以及欧拉函数的值。

子任务 4

$$\Pi_{i\le n}f(i)=k^{k*\sum_{i\le n}i*[i是素数]}*\left(\Pi_{i\le n}F(i)\right)^k$$

$$\Pi_{i\le n}g(i)=k^{k*\sum_{i\le n}\varphi(i)}*\left(\Pi_{i\le n}G(i)\right)^k$$

于是可以将$\sum_{i\le n}i*[i是素数]$和$\sum_{i\le n}\varphi(i)$的结果也打表存下，然后转化成$k=1$的情况，套用上一个子任务的解法即可。

子任务 5

容易发现时间复杂度的消耗主要在于计算第 4 个式子，因为计算 $\Pi_{i=l\sim r}F(i)$可以只有当 i 为素数时才需要进行一次快速幂，而$\Pi_{i=l\sim r}G(i)$则对于每个 i 都要进行一次快速幂，所以通过特判跳过关于第 4 个式子的所有计算即可。

子任务 6

$\sum_{i\le n}i*[i是素数]$和$\sum_{i\le n}\varphi(i)$可以用数论方法计算而无需打表(如 min_25 筛法，杜教筛)，于是可以把表的储存信息翻倍，同时还可以把数字用字符进行压缩，这样表的储存信息几乎又可以翻倍，这样就足以通过该子任务了。

以上做法十分暴力，接下来将介绍一个不那么暴力的做法。

子任务 7&8

先只考虑计算$\Pi_{i\le n}F(i)$，类似 min_25 筛法来设状态：

$$A(x)=\sum_{i\le x} i*[i\text{是素数或}i\text{的最小质因子}>p],$$

$$B(x)=\Pi_{i\le x} i^{i*[i\text{是素数或}i\text{的最小质因子}>p]}$$

对于每个素数 p，按 x 从大到小更新，则对于$\mathrm{x}\ge \mathrm{p}^2$

$$B(x)*=\left(B\left(\left\lfloor\frac{x}{p}\right\rfloor\right)^{p}*p^{A\left(\left\lfloor\frac{x}{p}\right\rfloor\right)}\right)^{-1}*B(p-1)^{p}*p^{A(p-1)}$$

由于多了快速幂，这样的总复杂度是$O\left(n^{\frac{3}{4}}\right)$的。

*有一个十分有效的优化，注意$\left\lfloor\frac{x}{p}\right\rfloor$的取值只有$O(\sqrt{\frac{n}{p}})$种，对于相同的值没有必要重新计算。

现在还有一个很大的问题，那就是初值$B(x)=\Pi_{i\le x} i^{i}$，这并不能 O(1)计算。

我们可以通过打表来解决，但直接将$\Pi_{i\le x} i^{i}$的结果打表效率很低，每次计算一个区间$[l,r]$需要花费$O((r-l)*\log n)$的代价，考虑将其进行一个转化。

$$\Pi_{i\le x} i^{i}=\Pi_{i\le x}\left(\frac{i\,!}{(i-1)\,!}\right)^{i}=(\Pi_{i<x} i!)^{-1}*x!^{x}$$

这样只需要将阶乘和阶乘的前缀积打表即可，计算一个区间$[l,r]$代价为$O((r-l)+\log n)$。

对于计算$\Pi_{i\le n}G(i)$，我们可以考虑对每个素数 p 独立计算贡献

$$\Pi_{i\le n}G(i)=\Pi_{p\le n}p^{(p-1)*\sum_{p^e\le n} e*p^{e-1}*\sum_{i\le\lfloor\frac{n}{p^e}\rfloor}\varphi(i)*[gcd(i,p)=1]}$$

设$S(n)=\sum_{i\le n}\varphi(i)\,,S_p(n)=\sum_{i\le n}\varphi(i)*[gcd(i,p)=1]$

则$S_p(n)=S(n)-(S_p\left(\left\lfloor\frac{n}{p}\right\rfloor\right)*(p-1)+(S\left(\left\lfloor\frac{n}{p}\right\rfloor\right)-S_p\left(\left\lfloor\frac{n}{p}\right\rfloor\right))*p)$

即$S_p(n)=S(n)-S\left(\left\lfloor\frac{n}{p}\right\rfloor\right)*p+S_p\left(\left\lfloor\frac{n}{p}\right\rfloor\right)$

$$\Pi_{i\le n}G(i)=\Pi_{p\le n}p^{(p-1)*\sum_{p^e\le n} e*p^{e-1}*S_p(\left\lfloor\frac{n}{p^e}\right\rfloor)}$$

考虑将 p 分成$\le\sqrt{\mathrm{n}}$和$>\sqrt{\mathrm{n}}$两类

对于 p≤ $\sqrt{\mathrm{n}}$，可以暴力枚举计算。

对于 p> $\sqrt{\mathrm{n}}$, $\Pi_{\sqrt{n}<p\le n}p^{(p-1)*S_p\left(\left\lfloor\frac{n}{p}\right\rfloor\right)}=\Pi_{\sqrt{n}<p\le n}p^{(p-1)*S\left(\left\lfloor\frac{n}{p}\right\rfloor\right)}$

可以分块计算，问题在于如何计算$\Pi_{p\le n}p^{(p-1)}$。

而当前已经能够计算$\Pi_{p\le n}p^{p}$，于是问题可以转变为计算$\Pi_{p\le n}p$，这是一个弱化版的问题，类似的设状态即可计算。

现在单次计算$\Pi_{i\le n}G(i)$可以做到$O\left(\sqrt{n}*\log p\right)$，其中的$\log p$来自于快速幂的复杂度，而事实上可以做到单次$O\left(\sqrt{n}\right)$，因为所有要进行快速幂的底数只有$O\left(\sqrt{n}\right)$种，我们可以将指数累加记录下来，最后每种底数只需要进行一次快速幂。

类似杜教筛，我们还可以设一个阈值m，小于阈值的部分预处理，大于的部分再调用上述做法。

m最优时有$m\log n=\sqrt{n}*\sqrt{\frac{n}{m}}$，则$m=\left(\frac{n}{\log n}\right)^{\frac{2}{3}}$,此时复杂度为$O(n^{\frac{2}{3}}*\log^{\frac{1}{3}}n)$

忽略打表部分的复杂度，最终总复杂度为$O\left(n^{\frac{3}{4}}+n^{\frac{2}{3}}*\log^{\frac{1}{3}}n+\sqrt{n}*\log p\right)$。