目录

1 A

先算烷基, 即有根树并且根的度数 ≤ 3.

设 $A(x)$ 为烷基的个数的生成函数. 根据 Pólya 定理, 我们有

$$A(x) = 1 + x\frac{A(x)^3 + 3A(x)A(x^2) + 2A(x^3)}{6}$$

这个可以用分治 FFT $O(n\log^2 n)$ 解, 需要用到一些技巧.

考虑烷烃个数的生成函数 $B(x)$. 但是并不方便直接用 $A(x)$ 表示出 $B(x)$.

对于一棵无根树, 令 p 和 q 分别表示这棵树的点等价类个数和边等价类个数. 定义对称边为满足连接的两个点是等价的的边 (显然这种边最多只有 1 条). 令 s 表示对称边的个数.

那么有下面这个式子恒成立:

$$p - q + s = 1$$

证明十分简单. $s = 0$ 时, 选任意一个重心做根, 容易证明没有其它点与这个根等价; 然后再考虑每个点及其父边的贡献即可. $s = 1$ 时的情况还更简单一些.

有了这个式子, 接下来的事情就好办了. 对于所有 n 个点的烷烃, 有:

$$\sum p - \sum q + \sum s = \sum 1$$

右边就是我们要求的.

令 $P(x)$ 表示烷烃的 $\sum p$ 的生成函数. 对于一个无根树, 选 n 个点中的任意一个点做根形成互不同构的的有根树的数量就是 p.

用一下 Pólya 定理, 我们有

$$P(x) = x\frac{A(x^4) + 3A(x^2)^2 + 6A(x)^2A(x^2) + 8A(x)A(x^3) + 6A(x^4)}{24}$$

再令 $Q(x)$ 表示烷烃的 $\sum q$ 的生成函数. 对于一个无根树, 选 $n-1$ 条边中的任意一条边劈开, 插入一个度数为 2 的点形成的互不同构的有根树的数量就是 q.

类似地, 我们有

$$Q(x) = \frac{(A(x)-1)^2 + (A(x^2)-1)}{2}$$

然后显然 $\sum s$ 的生成函数就是 $A(x^2)$.

所以最终烷烃的数量的生成函数为

$$B(x) = P(x) - Q(x) + A(x^2)$$

时间复杂度为 $O(n\log^2 n + T)$.

2 B

考虑 Burnside 引理.

如果没有颜色的变换, 那么是一道很经典的题, 共 $2n$ 个置换. 那么我们可以将颜色变换这一种看似不是置换的考虑进置换群里面去, 现在变成了 $2nm$ 个置换.

考虑如何快速计算带有颜色取反的置换的不动点个数.

对于旋转的置换:

$$
\begin{aligned}
&\sum_{d=1}^{n} m^{\gcd(d,n)} \sum_{g=1}^{m} \left[\frac{m}{\gcd(g,m)} \mid \frac{n}{\gcd(d,n)}\right] \\
=&\sum_{d\mid n} \gcd(d,m)\varphi(d) m^{\frac{n}{d}}
\end{aligned}
$$

翻转的置换是类似的.

注意这题需要 Pollard's Rho 来分解质因数.

3 C

莫比乌斯反演. 原式化为:

$$\sum_{i=1}^{A}\sum_{j=1}^{B}\sum_{k=1}^{C}\sum_{d|i,d|j^2,d|k^3}(\mu*\varphi)(d)$$

调整, 得:

$$\sum_{d=1}^{A}(\mu*\varphi)(d)\sum_{i,d|i}^{A}\sum_{j,d|j^2}^{B}\sum_{k,d|k^3}^{C}1$$

令 $x=\prod_i p_i^{a_i}$ (质因数分解), 注意到满足 $x|y^k$ 当且仅当 $(\prod_i p_i^{\lceil\frac{a_i}{k}\rceil})|y$.
记 $f_k(x)=\prod_i p_i^{\lceil\frac{a_i}{k}\rceil}$, 原式化为:

$$\sum_{d=1}^{A}(\mu*\varphi)(d)\lfloor\frac{A}{f_1(d)}\rfloor\lfloor\frac{B}{f_2(d)}\rfloor\lfloor\frac{C}{f_3(d)}\rfloor$$

$\mu*\varphi$ 和 $f_k(x)$ 显然具有积性, 因此可以线性筛.
由此, 本题的复杂度可以做到 $O(A)$.

4 D

考虑一个 n 个点的图, 其中点 i 向 p_i 连边, 显然这张图是由若干个环组成的, 并且排列的权值就是每条边的长度之和.

考虑从前往后 DP. 设状态 $dp(i,j,k)$ 表示考虑了前 i 个位置, 有 j 条出边连出, j 条入边连入 $1\ldots i$ 这段前缀, 且当前的权值为 k. 转移十分显然.

时间复杂度 $O(n^4)$.

5 E

你要是问我 TeaTree 是谁, 当然是 C 菌的好基友茶理理啦…

这题解法很粗暴, 3 秒时限 1G 内存, 不粗暴才怪了.

小于 100000 的数最多只有一百多个约数.

建出所有点的约数线段树, 然后线段树合并, 重复的单点就可以对答案产生贡献.

当然有省内存的做法, DSU on tree, 不过反正时间复杂度不变.

时间与空间都是 $O(100 \times n \log n)$

6 F

首先题目就是要你求每对点之间的最大流对 3 取 min 的和. 也就是要判断

- 有多少对点满足存在一种方案, 删去一条边后不连通
- 有多少对点满足存在一种方案, 删去两条边后不连通

第一种很好算, 找出边双连通分量即可. 不妨假设现在整张图是边双连通的.

先搞出 DFS 生成树来, 容易发现删去的两条边可能是:

某一条边是树边, 另一条边是非树边 要求这条树边仅被这条非树边覆盖

两条边都是树边 要求这两条树边被覆盖情况相同

不妨考虑类似 共价大爷游长沙 的思路, 即, 给每条非树边随机一个 10^{18} 以内的权值, 一个树边的权值为覆盖它的非树边的权值的异或和. 然后用上 `map` 就很好统计了.

时间复杂度 $O(n \log m)$.

7 G

考虑使用容斥原理进行计数.

包含至少一个形如 $[i, i+1]$ 或 $[n, 1]$ 这样的子串的环排列个数是 $\binom{n}{1}(n-2)!$ 个;

可以推广为包含至少 $k(0 \le k < n)$ 个的环排列个数是 $\binom{n}{k}(n-k-1)!$, 同时注意到包含 n 个的环排列个数一定是 1 个.

所以最终答案就是

$$(-1)^n + \sum_{k=0}^{n-1} (-1)^k \binom{n}{k} (n-k-1)!$$

8 H

```
printf("%.0f\n", pow(2, n));
```

9 I

令 $a=i-j$, 先枚举 i 再枚举 a

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{i-1}[\gcd(i+j,i-j)=1]\\
=&\sum_{i=1}^{n}\sum_{a=1}^{i-1}[\gcd(2i-a,a)=1]\\
=&\sum_{i=1}^{n}\sum_{a=1}^{i-1}[\gcd(2i,a)=1]
\end{aligned}$$

即对于每个 i, 求有多少个小于它的 a 满足 $\gcd(i,a)=1$ 且 a 是奇数.
当 i 是奇数时, 答案为 $\frac{\varphi(i)}{2}$.
当 i 是偶数时, 答案为 $\varphi(i)$.
注意 $i=1$ 时, 答案为 0.
记个前缀和就好了, 复杂度为 $O(N+T)$.
另一种有趣的解法, 设

$$\begin{aligned}
f(n)&=\sum_{i=1}^{n}\varphi(i)\\
g(n)&=\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}\varphi(2i)=\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}\varphi(i)+\sum_{i=1}^{\lfloor\frac{n}{4}\rfloor}\varphi(2i)=f(\lfloor\frac{n}{2}\rfloor)+g(\lfloor\frac{n}{2}\rfloor)
\end{aligned}$$

则

$$\begin{aligned}
Ans&=\frac{f(n)-g(n)-1}{2}+g(n)\\
&=\frac{f(n)+g(n)-1}{2}\\
&=\frac{f(n)+f(\lfloor\frac{n}{2}\rfloor)+g(\lfloor\frac{n}{2}\rfloor)-1}{2}\\
&=\frac{f(n)+f(\lfloor\frac{n}{2}\rfloor)+f(\lfloor\frac{n}{4}\rfloor)+g(\lfloor\frac{n}{4}\rfloor)-1}{2}\\
&\vdots\\
&=\frac{\sum_{i=0}^{\infty}f(\lfloor\frac{n}{2^i}\rfloor)-1}{2}
\end{aligned}$$

虽然单次求解复杂度多了一个 log, 但是配合上杜教筛就可以解决 10^9 的范围了, 唯一的缺点是这样就不支持多组询问, 所以没出到题目里面.

10 J

K 不大于 5, 仅是常数级别, 所以可以搞事情

我们发现 $|x_{MW}[i] - x_{SW}[i]| = max(x_{MW}[i] - x_{SW}[i], x_{SW}[i] - x_{MW}[i])$

也就是说对于每一个维度只有两种选择, 同时 $2^K \leq 32$ 也不大, 所以可以枚举每一维的大小情况, 分别取主武器与副武器的最大值就好了, 复杂度 $O(2^K n)$.

11 K

SillyDarkGK 这个名字怎么这么中二非主流啊？其实是指 B 站三怂之首蠢黑少爷啦…

令 $S = A - B$ 则答案为 $calc_{add}(A) + calc_{dec}(B)$, 由于某些位数不能放数字, 所以如果想要在这一位凑出 1, 需要用很多次低位的数字, 总之我们可以求出在每一位加或者减需要的代价.

我们可以以加或者减的方式填充, 以加的方式填充很简单, 减则是

```
1000000000000 (A)
-001010000001 (B)
=110101111111 (S)
```

我们可以看出以加的方式填充需要的代价就是 A=S 的代价, 以减的方式填充则主要是 B=~S 的代价

设 dp[i][0] ——填充了前 i 小的位, 目前以加的方式填充的最小代价

设 dp[i][1] ——填充了前 i 小的位, 目前以减的方式填充的最小代价

dp[i-1][0] 转 dp[i][0] 与 dp[i-1][1] 转 dp[i][1] 都很简单

dp[i-1][0] 转 dp[i][1] 需要多付出 B[i] 的代价, 明显 S 在第 i 位要是 1, 否则不优

dp[i-1][1] 转 dp[i][0] 需要多付出 A[i] 的代价, 明显 S 在第 i 位要是 0, 否则不能转

具体 DP 方式见 std

12 L

C-bacteria 是谁啊？我是指 C 菌…

讲讲我出本题的心路历程吧, 假设题目中的视频没有种类的说法 (自然不会有连续观看同种视频掉快乐度), 那么我们有一个经典最大费用流套路可以用

先建出主链 (括号里的二元组表示边权, 第一个数字是流量, 第二个是费用)

```
 (K,0) (inf,0)(inf,0)      (inf,0)(inf,0)
S----->o----->o----->o ... ----->o----->T
        时间1    时间2  时间3          时间4
```

S 与 T 之间的每一个节点表示时间, 每一个从 st 开始到 ed 结束权为 w 的视频则是一条从时间 st‘连向 ‘时间 ed‘的权值为 ‘(1,w)‘的边

这个套路的思想是，用流量来模拟人，在主链上的人表示在休息，如果一条流量经过视频边则表示此人观看此视频

其实我原计划是把题目出成这样

有两座城堡，城堡 A 与城堡 B，初始驻军量分别为 a 支军队与 b 支军队，有 m 件任务。每件任务形式如：A 城堡在 i 时间派出一支军队出城搜查，这支军队会在 j 时间回到 B 城堡（或者 B 城堡去 A 城堡，保证 i 小于 j），如果选择执行这一任务会获利 val，选择不执行任务则不会有军队调动，求最大收益

但是这样好像就太明显了吧…

建立两条主链，分别表示城堡 A 与城堡 B 的时间轴，任务就是从 A 轴某时间向 B 轴某时间连边（或反过来）

于是乎本题的做法也水落石出了，相比于原题目，本题只需要多加上一条从 A 轴向 A 轴（或 B 向 B）的权值为 ‘(1,w-W)‘连边就好了