

WC2019 模拟题

nqiiii

2018 年 12 月 19 日

1 操作

每次操作可以看作把 x 的期望代入了一个一次函数 $D_i x + E_i$ 。我们计算每个 E_i 期望被乘上多少。对于第 i 个操作，我们对于所有 $j \neq i$ ，把 $1 + D_j x$ 乘起来，得到的多项式的 x^k 项的系数再乘上 $k!(n-1-k)!$ 即为有 k 个操作排在第 i 个操作后面的所有方案乘上的数之和。

我们需要对于每个 i 求出对应的多项式乘上 E_i 的值，在把这些多项式求和。这个过程显然可以使用分治 NTT 优化。

时间复杂度 $O(n \log^2 n)$ 。

2 选数

我们设 m 为值域， $f(i)$ 为选出的所有数都为 i 的倍数的方案数，经过简单的莫比乌斯反演，答案就是 $\sum_{i=1}^m \varphi(i) f(i)$ 。

我们对于每个 $j \leq k$ 求出可以重复选 j 个数的方案数，然后通过容斥就可以求出不能重复选 k 个的方案数。我们设一个阈值 L ，如果 $i \leq L$ ，那么我们使用快速沃尔什变换求 $f(i)$ ，复杂度为 $O(m \log m)$ 。如果 $i > L$ ，我们用 meet in the middle 求 $f(i)$ ，复杂度为 $O(\lfloor \frac{m}{i} \rfloor^2)$ 。

如果我们设 $L = \sqrt{m}$ ，那么复杂度为 $O(m\sqrt{m} \log m)$ 。

3 字符串

如果我们把所有长为 $k-1$ 的串看作一张图上的点，每一个长为 k 的串看作从它的长为 $k-1$ 的前缀连向它长为 $k-1$ 的后缀的边，那么一个字符串可以看作这张图上的一条路径。我们要找到一条路径，满足所有 wzz 串代表的的路径必须被经过至少一次，并且长度最短。

如果我们把没被经过的边都删掉，一条边被经过两次就再连一条同样的边，那么这样得到的图一定存在一条欧拉路径。那么我们要做的就是先连上所有 wzz 串代表的边，要求加最少的边使得这张图有一条欧拉路径。

首先，只有所有 wzz 串代表的边的图一定是除孤立点外连通的，因为 $k \times 2 \leq |S|$ 。定义一个点的度数为入度减出度，那么只要满足除了两个点度数为 1 和 -1，其他点度数都为 0 即可。我们考虑这样一个 fake 网络流问题，我们把所有度数为大于 0 的点看作源点集合，流出的流量为度数，把所有度数小于 0 的点看作汇点集合，流入的流量为度数的相反数，然后我们把每条可能存在的边的容量看成 inf，费用看成 1，我们就是需要找到一种方案，使得最多一个源点实际流出量比应有流出量少 1，最多一个汇点实际流入量比应有流入量少 1，然后要求费用最小。

我们使用一个贪心策略，每次找到所有源点到汇点的路径中最短的一条，然后增广这条路径（不使用反向边），直到满足条件为止。接下来我们证明这样做是对的。

如果当前的最短路是从点 u 到点 x 的一条路径，假设存在一组最优解中不包含这条路径。

如果在这组解中点 u 向外没有流量并且没有流量流向点 x ，那么显然不平衡的点是点 u 和点 x ，随便把另一条路径换成这条即可。

如果在这组解中点 u 向外没有流量，那么随便找一条到点 x 的路径换成这条即可。

如果在这组解中没有流量流向点 x ，那么随便找一条从点 u 出发的路径换成这条即可。

对于最后一种情况，我们假设在这组解中有一条点 u 到点 y 的路径和一条点 v 到点 x 的路径，那么我们证明把它换成点 u 到点 x 路径和点 v 到 y 的路径也是可以的。即我们要证 $\text{dis}(u, x) + \text{dis}(v, y) \leq \text{dis}(u, y) + \text{dis}(v, x)$ 。

由于这张图上的每个节点都代表着一个长为 $k - 1$ 的字符串，所以如果 $A(p, q)$ 为最长的字符串，满足它是 p 的后缀和 q 的前缀，那么 $\text{dis}(p, q) = k - 1 - |A(p, q)|$ 。那么我们要证的变成了 $|A(u, x)| + |A(v, y)| \geq |A(u, y)| + |A(v, x)|$ 。

我们设 $t = |A(u, y)| + |A(v, x)| - |A(u, x)|$ ，那么就是要证 $|A(v, y)| \geq t$ 。如果 $t \leq 0$ ，显然是对的。否则因为 $\text{dis}(u, x)$ 是最小的，那么 $A(u, y)$ 就是 $A(u, x)$ 的前缀， $A(v, x)$ 就是 $A(u, x)$ 的后缀。则一定存在一个字符串 z ，满足 $|z| = t$ 且它是 $A(v, x)$ 的后缀和 $A(u, y)$ 的前缀。那么它一定是 v 的后缀和 y 的前缀。所以 $|A(v, y)| \geq |z| = t$ 。

接下来我们模拟这个做法。我们把所有 wzz 串的长为 $k - 1$ 的前缀丢到一个可重集合 P 中，把长为 $k - 1$ 的后缀丢到一个可重集合 S 中。初始的代价是 0。我们每次去除 P 和 S 中重复的元素，每去除一次把答案加上当前的代价。之后我们删除 P 中所有串的最后一个字符和 S 中所有串的第一个字符，把代价加 1，然后重复以上过程直到 P 和 S 为空。容易发现一次去重就等于一次增广，而去重的代价就是增广的费用。由于这种方法多增广了一次，那么我们把答案减去最后一次去重的代价即可。由于 P 和 S 中的元素可能会很多，但我们发现如果两个串把数字替换成字母后一样，那么这两个串的出现次数也一样，我们在一起统计即可。

时间复杂度 $O(nk \log k)$ 。