题目选讲

ztzshiwo

2018 年 1 月 9 日

热身题[1]

众所周知, 萌萌哒六花不擅长数学, 所以勇太给了她一些数学问题做练习.

其中有一道是这样的:

对于 n 个 01 字符串 s_i, 定义他们的权值是这 n 个 s_i 的串插入一个空的 $Trie$ 树后得到的结果 $Trie$ 中的节点个数. 例如 $[”01”, ”00”]$ 的权值是 4,$[”010”, ”1”]$ 的权值是 5.

现在勇太给出了 n 个只包含 $01?$ 的字符串 s_i. 其中? 表示既有可能是 0 也有可能是 1. 显然, 如果有 K 个?, 那么一共有 2^K 个可能的字符串集合.

现在勇太想要知道对于这 2^K 种状态, 权值的和是多少.

当然, 这个问题对于萌萌哒六花来说实在是太难了, 你可以帮帮她吗？

$1 \leq n \leq 20, 1 \leq |s_i| \leq 50$, 对 998244353 取模

[1]hihocoder#1646 : Rikka with String II

容斥原理

- 看到 $n \le 20$ 的数据范围，显然可以想到容斥原理.

容斥原理

- 看到 $n \leq 20$ 的数据范围，显然可以想到容斥原理.
- 可以发现 $Trie$ 树的节点数为

$$1 + \sum_{S} (-1)^{|S|-1} lcp(\sum_{s_i \in S})$$

容斥原理

- 看到 $n \leq 20$ 的数据范围，显然可以想到容斥原理.
- 可以发现 $Trie$ 树的节点数为

$$1+\sum_{S}(-1)^{|S|-1}lcp(\sum_{s_i \in S})$$

- 观察式子以后可以发现正负性只和集合中的元素数量有关.

  可以枚举长度分别计算满足当前条件的方案数，然后乘上容斥系数加在一起.

容斥原理

- 看到 $n \leq 20$ 的数据范围，显然可以想到容斥原理.
- 可以发现 $Trie$ 树的节点数为

$$1 + \sum_{S} (-1)^{|S|-1} lcp(\sum_{s_i \in S})$$

- 观察式子以后可以发现正负性只和集合中的元素数量有关.
  可以枚举长度分别计算满足当前条件的方案数，然后乘上容斥系数加在一起.
- 这样的复杂度是 $O(|s_i| n 2^n)$ 的. 常数小一点的话是可以过的.

小优化

- 我们对于每一个状态存储到达当前状态的方案数.
- 计算状态时只要取出最后一个与前面计算就可以了.
- 具体计算的时候记录两个数组 ans, f 分别代表到达当前状态的方案数以及相同所需的限制
- 总复杂度为 $O(|s_i|2^n)$, 可以通过此题.

套路题[2]

众所周知，萌萌哒六花不擅长数学，所以勇太给了她一些数学问题做练习.

其中有一道是这样的:

勇太有一棵 n 个点的树，现在他想要将这棵树的每一个节点染色。
染色方案必须满足树上每一个节点数不超过 K 的联通子图中的所有节点颜色不同。
现在勇太想要知道他至少需要多少种不同的颜色才能对这棵树完成染色。

当然，这个问题对于萌萌哒六花来说实在是太难了，你可以帮帮她吗?

$1 \leq K \leq n \leq 10^5$

[2] hihocoder#1462 : Rikka with Tree IV

考虑包含 k 个点的链，可以得出任意距离不超过 $k-1$ 的点的颜色不同。

在大小为 k 的连通子图中，任意两点的距离不超过 $k-1$，所以这个条件是充要的。

考虑包含 k 个点的链，可以得出任意距离不超过 $k-1$ 的点的颜色不同。

在大小为 k 的连通子图中，任意两点的距离不超过 $k-1$，所以这个条件是充要的。

考虑 $K=2$ 的情况，就是任意两个相邻的节点不能相同颜色。

$K=3$ 就是每个点和它的邻居的颜色两两不同.

考虑包含 k 个点的链，可以得出任意距离不超过 $k-1$ 的点的颜色不同。

在大小为 k 的连通子图中，任意两点的距离不超过 $k-1$，所以这个条件是充要的。

考虑 $K=2$ 的情况，就是任意两个相邻的节点不能相同颜色。

$K=3$ 就是每个点和它的邻居的颜色两两不同.

我们可以枚举任何一条链的中心点，与这个点距离不超过 $\frac{k-1}{2}$ 的点都可以任意形成一个长度 $\leq k$ 的链。

我们对于每一个点求出距离它不超过 $\frac{k-1}{2}$ 的点的数量，然后取 MAX 就可以了。

对于偶数的情况，就把边作为链的中心。暴力的复杂度为 $O(n^2)$

我们对于每一个点求出距离它不超过 $\frac{k-1}{2}$ 的点的数量，然后取 MAX 就可以了。

对于偶数的情况，就把边作为链的中心。暴力的复杂度为 $O(n^2)$

如果通过点分治计算，就可以达到 $nlogn$ 的复杂度.

套路题[3]

在一个 s 个点的图中，存在 $s-n$ 条边，使图中形成了 n 个连通块.
第 i 个连通块中有 a_i 个点。

现在我们需要再连接 $n-1$ 条边，使该图变成一棵树.
对一种连边方案，设原图中第 i 个连通块连出了 d_i 条边.

那么这棵树 T 的价值为:

$$val(T)=\left(\prod_{i=1}^{n} d_i{}^m\right)\left(\sum_{i=1}^{n} d_i{}^m\right)$$

你的任务是求出所有可能的生成树的价值之和，对 998244353 取模。

$n \le 3\times 10^4, m \le 30$

[3] LOJ2320 「清华集训 2017」生成树计数

列出初始式子

可以根据题意推出式子

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \frac{(n-2)!}{\prod_{i=1}^{n} d_i!} \prod_{i=1}^{n} a_i^{d_i+1} \prod_{i=1}^{n} (d_i+1)^m \sum_{i=1}^{n} (d_i+1)^m$$

首先可以考虑 $m=1$ 的情况，可知 $\sum_{i=1}^{n}(d_i+1)=2n-2$

式子推导方式与之前提到的一道 dp 题很像。

列出初始式子

可以根据题意推出式子

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \frac{(n-2)!}{\prod_{i=1}^{n} d_i!} \prod_{i=1}^{n} a_i^{d_i+1} \prod_{i=1}^{n} (d_i+1)^m \sum_{i=1}^{n} (d_i+1)^m$$

首先可以考虑 $m=1$ 的情况，可知 $\sum_{i=1}^{n}(d_i+1)=2n-2$

式子推导方式与之前提到的一道 dp 题很像。

式子变为

$$(2n-2)(n-2)! \prod_{i=1}^{n} a_i \sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i}(d_i+1)}{d_i!}$$

运用套路

前面的都很好计算，只需要看 $\sum$ 后面的式子，考虑把 d_i+1 拆开

$$\sum_{\left(\sum_{i=0}^{n} d_i\right)=n-2} \sum_{S} d_i^{i\in S} \prod_{i=1}^{n} \frac{a_i^{d_i}}{d_i!}$$

运用套路

前面的都很好计算，只需要看 $\sum$ 后面的式子，考虑把 $d_i + 1$ 拆开

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \sum_{S} d_i^{i\in S} \prod_{i=1}^{n} \frac{a_i^{d_i}}{d_i!}$$

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \sum_{S} a_i^{i\in S} \prod_{i=1}^{n} \frac{a_i^{d_i-[i\in S]}}{(d_i-[i\in S])!}$$

运用套路

前面的都很好计算，只需要看 $\sum$ 后面的式子，考虑把 d_i+1 拆开

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \sum_{S} d_i^{i\in S} \prod_{i=1}^{n} \frac{a_i^{d_i}}{d_i!}$$

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \sum_{S} a_i^{i\in S} \prod_{i=1}^{n} \frac{a_i^{d_i-[i\in S]}}{(d_i-[i\in S])!}$$

$$\sum_{S} a_i^{[i\in S]} \sum_{(\sum_{i=0}^{n} d_i)=n-|S|-2} \prod_{i=1}^{n} \frac{a_i^{d_i}}{d_i!}$$

运用套路

前面的都很好计算，只需要看 $\sum$ 后面的式子，考虑把 d_i+1 拆开

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \sum_S d_i^{i\in S} \prod_{i=1}^{n} \frac{a_i^{d_i}}{d_i!}$$

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \sum_S a_i^{i\in S} \prod_{i=1}^{n} \frac{a_i^{d_i-[i\in S]}}{(d_i-[i\in S])!}$$

$$\sum_S a_i^{[i\in S]} \sum_{(\sum_{i=0}^{n} d_i)=n-|S|-2} \prod_{i=1}^{n} \frac{a_i^{d_i}}{d_i!}$$

$$\sum_S a_i^{[i\in S]} \frac{(\sum_{i=1}^{n} a_i)^{n-|S|-2}}{(n-|S|-2)!}$$

运用套路

前面的都很好计算，只需要看 $\sum$ 后面的式子，考虑把 d_i+1 拆开

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \sum_S d_i^{i\in S} \prod_{i=1}^{n} \frac{a_i^{d_i}}{d_i!}$$

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \sum_S a_i^{i\in S} \prod_{i=1}^{n} \frac{a_i^{d_i-[i\in S]}}{(d_i-[i\in S])!}$$

$$\sum_S a_i^{[i\in S]} \sum_{(\sum_{i=0}^{n} d_i)=n-|S|-2} \prod_{i=1}^{n} \frac{a_i^{d_i}}{d_i!}$$

$$\sum_S a_i^{[i\in S]} \frac{(\sum_{i=1}^{n} a_i)^{n-|S|-2}}{(n-|S|-2)!}$$

可以发现实际上就是多项式的乘积

$$\left(\sum_{i=1}^{n} a_i x+1\right)\left(\sum_{i=0}^{n} \frac{(\sum_{j=1}^{n} a_j)^i}{i!} x^i\right)$$

求得这个多项式的第 $n-2$ 项系数再乘以之前的常数就可以得到答案

$m \le 30$

我们可不可以考虑将这个式子同样化为类似 $m=1$ 的形式呢？

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \frac{(n-2)!}{\prod_{i=1}^{n} d_i!} \prod_{i=1}^{n} a_i^{d_i+1} \prod_{i=1}^{n} (d_i+1)^m \sum_{i=1}^{n} (d_i+1)^m$$

$$(n-2)! \prod_{i=1}^{n} a^i \sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i}(d_i+1)^m}{d_i!} \sum_{i=1}^{n} (d_i+1)^m$$

后面的 $\sum$ 其实就是将其中一个 $(d_i+1)^m$ 变为 $(d_i+1)^{2m}$ ，我们可以先不管。

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i}(d_i+1)^m}{d_i!}$$

$m \leq 30$

我们可不可以考虑将这个式子同样化为类似 $m = 1$ 的形式呢？

$$\sum_{(\sum_{i=0}^{n} d_i) = n-2} \frac{(n-2)!}{\prod_{i=1}^{n} d_i!} \prod_{i=1}^{n} a_i^{d_i+1} \prod_{i=1}^{n} (d_i+1)^m \sum_{i=1}^{n} (d_i+1)^m$$

$$(n-2)! \prod_{i=1}^{n} a^i \sum_{(\sum_{i=0}^{n} d_i) = n-2} \prod_{i=1}^{n} \frac{a_i^{d_i}(d_i+1)^m}{d_i!} \sum_{i=1}^{n} (d_i+1)^m$$

后面的 $\sum$ 其实就是将其中一个 $(d_i+1)^m$ 变为 $(d_i+1)^{2m}$ ，我们可以先不管。

$$\sum_{(\sum_{i=0}^{n} d_i) = n-2} \prod_{i=1}^{n} \frac{a_i^{d_i}(d_i+1)^m}{d_i!}$$

我们可以用斯特林数代替多次幂

$$\sum_{(\sum_{i=0}^{n} d_i) = n-2} \prod_{i=1}^{n} \frac{a_i^{d_i} \sum_{j=0}^{m} C_{d_i+1}^{j} j! S(m,j)}{d_i!}$$

推式子

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i} \sum_{j=0}^{m} (C_{d_i}^{j} + C_{d_i}^{j-1}) j! S(m,j)}{d_i!}$$

推式子

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i} \sum_{j=0}^{m} (C_{d_i}^{j} + C_{d_i}^{j-1}) j! S(m,j)}{d_i!}$$

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i} (\sum_{j=0}^{m} C_{d_i}^{j} (j! S(m,j) + (j+1)! S(m,j+1)))}{d_i!}$$

推式子

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i} \sum_{j=0}^{m} (C_{d_i}^{j} + C_{d_i}^{j-1}) j! S(m,j)}{d_i!}$$

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i} (\sum_{j=0}^{m} C_{d_i}^{j} (j! S(m,j) + (j+1)! S(m,j+1)))}{d_i!}$$

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i-j} (\sum_{j=0}^{m} \frac{a_i^j}{j!} (j! S(m,j) + (j+1)! S(m,j+1)))}{(d_i-j)!}$$

推式子

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i} \sum_{j=0}^{m} (C_{d_i}^{j} + C_{d_i}^{j-1}) j! S(m,j)}{d_i!}$$

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i} (\sum_{j=0}^{m} C_{d_i}^{j} (j! S(m,j) + (j+1)! S(m,j+1)))}{d_i!}$$

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} \prod_{i=1}^{n} \frac{a_i^{d_i-j} (\sum_{j=0}^{m} \frac{a_i^j}{j!} (j! S(m,j) + (j+1)! S(m,j+1)))}{(d_i-j)!}$$

$$\sum_{(\sum_{i=0}^{n} d_i)=n-2} (\sum_{j=0}^{m} \frac{a_i^j}{j!} (j! S(m,j) + (j+1)! S(m,j+1))) \prod_{i=1}^{n} \frac{a_i^{d_i-j}}{(d_i-j)!}$$

多项式

由此可以看出式子化为了跟 $m=1$ 时相似的形式，也可以转化为多项式相乘

$$\left(\sum_{j=0}^{m} \frac{a_i^j(j!S(m,j)+(j+1)!S(m,j+1))}{j!}x^j\right)\left(\sum_{i=0}^{n} \frac{(\sum_{j=1}^{n} a_j)^i}{i!}x^i\right)$$

多项式

由此可以看出式子化为了跟 $m=1$ 时相似的形式，也可以转化为多项式相乘

$$\left(\sum_{j=0}^{m}\frac{a_i^j(j!S(m,j)+(j+1)!S(m,j+1))}{j!}x^j\right)\left(\sum_{i=0}^{n}\frac{(\sum_{j=1}^{n}a_j)^i}{i!}x^i\right)$$

考虑前面忽略的 $\sum$，只要把其中一个多项式变为 $2m$ 就行了。

总结为
设

$$A_i=\sum_{j=0}^{m}\frac{a_i^j(j!S(m,j)+(j+1)!S(m,j+1))}{j!}x^j$$为考虑第 i 个联通块度数的多项式（不变化）

$$B_i=\sum_{j=0}^{2m}\frac{a_i^j(j!S(2m,j)+(j+1)!S(2m,j+1))}{j!}x^j$$为考虑第 i 个联通块度数的多项式（变化）

F_i为考虑前 i 个联通块度数的多项式（其中没有联通块变化）

G_i为考虑前 i 个联通块度数的多项式（其中有联通块变化）

显然可以推出公式:

$$F_i=F_{i-1}*A_i$$

$$G_i=G_{i-1}*A_i+F_{i-1}*B_i$$

多项式

设

$$S=\sum_{i=1}^{n}a_i$$

$$C=\sum_{i=0}^{n}\frac{S^i}{i!}x^i$$

$$ans=(n-2)!\prod_{i=1}^{n}a_iG_n*C(n-2)$$

其中$G_n*C(n-2)$表示多项式$F=G_n*C$的第 n-2 项系数

G_n 可以通过分治 FFT 求，总复杂度 $O(nmlogn)$

这一次的讲题比较仓促，所以选的都是一些比较格式化的题.

没有什么巧妙的变换或者思路。

题解很大一部分也是从各种相关资料里面摘抄的。

在此表示歉意.

谢谢大家