NOI2016模拟赛

Newnode

June 16, 2016

1 deep

1.1 题目大意

n个点的树，有一些点是关键点，要求划分连通块使得每个连通块都包含至少一个关键点，最大的连通块最小。

1.2 100%算法

显然可以二分答案，假设现在二分的值为k，考虑如何判定。

我们设$f_{i,0/1}$为只考虑i的子树，i所在的连通块最小是多少（0和1表示包不包含关键点，显然如果可以包含关键点就一定让它包含，所以可以用f_i的正负性来代替记录两个量）。

我们可以贪心地求解f。

如果i是关键点，那么对于f为正（也就是根所在的连通块有关键点）的子树就不用包含在i的连通块中了，剩下的一定要在i中，直接累加就可以得到f_i。

如果i不是关键点，选择f为正且f最小的子树，让i和f为负的子树都包含在其中，如果大小超过k的话就只能让i和f为负的子树在一个连通块了，f_i就为负。

时间复杂度：$O(n\log n)$

2 dark

2.1 题目大意

n个点的无向图，每条边都可能存在，一个图的权值是连通块个数的m次方，求所有可能的图的权值和。

2.2 40%算法

设g_i表示i个点的连通图有多少个，$f_{i,j}$表示i个点j个连通块的图有多少个。我们可以通过枚举1号点所在连通块大小得到转移式

$$f_{i,j} = \sum_{k=1}^{i} f_{i-k,j-1} g_k \binom{i-1}{k-1}$$

考虑如何求g_i，设h_i表示i个点的图有多少个，显然

$$h_i = 2^{\frac{i(i-1)}{2}}$$

那么就可以通过经典的容斥——枚举1号点所在连通块大小得到转移式

$$g_i = h_i - \sum_{j=1}^{i-1} g_j h_{i-j} \binom{i-1}{j-1}$$

直接按照上述式子dp然后统计答案，时间复杂度$O(n^3 + nT)$。

2.3 60%算法

可以直接设$f_{i,j}$表示i个点的图，权值是连通块个数的m次方的权值和，同样枚举1号点所在的连通块大小来转移，注意到$(n+1)^m = \sum_{i=0}^{m} \binom{m}{i} n^i$，于是可以得到转移式

$$f_{i,j} = \sum_{k=1}^{i} g_k \binom{i-1}{k-1} \sum_{t=0}^{j} \binom{j}{t} f_{i-k,t}$$

注意到$\sum_{t=0}^{j} \binom{j}{t} f_{i-k,t}$和$i$没有直接联系，可以预处理一下所以不需要枚举$t$，时间复杂度$O(n^2 m + T)$。

2.4 100%算法

考虑第二类斯特林数$\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$的意义，是将$n$个不同的元素拆分成$m$个集合的方案数，而$n^m$是在$n$个元素中选择$m$次的方案数，我们可以枚举选了几个不同的元素$k$，那么就要将这$m$次拆成$k$个集合，所以有

$$n^m = \sum_{k=0}^{m} \left\{ \begin{matrix} m \\ k \end{matrix} \right\} \binom{n}{k} k!$$

利用这个式子，我们可以将原题中的连通块数^{m}转化为$\binom{\text{连通块数}}{m}$。

那么就可以设$F_{i,j}$表示i个点的图，权值是$\binom{\text{连通块数}}{m}$的权值和，同样枚举1号点所在的连通块大小来转移，注意到$\binom{n+1}{m}=\binom{n}{m}+\binom{n}{m-1}$，于是可以得到转移式

$$F_{i,j}=\sum_{k=1}^{i}g_k\binom{i-1}{k-1}(F_{i-k,j}+F_{i-k,j-1})$$

那么就有

$$f_{n,m}=\sum_{k=0}^{m}\left\{{m \atop k}\right\}F_{n,k}k!$$

而

$$\left\{{m+1 \atop k}\right\}=k\left\{{m \atop k}\right\}+\left\{{m \atop k-1}\right\}$$

所以可以预处理第二类斯特林数，每个询问$O(m)$计算答案，那么现在问题就在于如何高效求出F。

注意到m很小，而且$f_{i,j}$只和j与$j-1$层有关，于是可以分层做。

我们稍微化简一下F的转移式，得到

$$\frac{F_{i,j}}{(i-1)!}=\sum_{k=1}^{i}\frac{g_k}{(k-1)!}\frac{F_{i-k,j}+F_{i-k,j-1}}{(i-k)!}$$

于是可以分治FFT计算，我们用$Solve(l,r)$求出l到r的函数值，那么只需要先$Solve(l,\frac{l+r}{2})$，然后再用FFT算出这一部分对后一部分的贡献，再$Solve(\frac{l+r}{2}+1,r)$就好了。

而g也可以用类似的办法分治FFT求出。

时间复杂度：$O(mn\log^2 n+mT)$

2.5 100%算法2

我们考虑$\binom{\text{连通块数}}{m}m!$的组合意义——在所有的连通块中有序地选择m个。

那么可以设$H_{i,j}$表示i个点的图，共有j个连通块，有序选择j个的和，枚举这一次选的连通块可以发现

$$H_{i,j}=\sum_{k=1}^{i}g_k\binom{i}{k}H_{i-k,j-1}$$

也就是

$$\frac{H_{i,j}}{i!}=\sum_{k=1}^{i}\frac{g_k}{k!}\frac{H_{i-k,j-1}}{(i-k)!}$$

于是可以用m次FFT求出H。

考虑如何计算答案

$$f_{n,m}=\sum_{i=0}^{n}\sum_{j=0}^{m}\left\{{m \atop j}\right\}j!h_{n-i}\binom{n}{i}H_{i,j}$$

那么就可以设

$$G_{a,b}=\sum_{i=0}^{a}h_{a-i}\binom{a}{i}H_{i,b}$$

也就是

$$\frac{G_{a,b}}{a!}=\sum_{i=0}^{a}\frac{H_{i,b}}{i!}\frac{h_{a-i}}{(a-i)!}$$

于是可以用m次FFT求出G。

那么答案就是

$$f_{n,m}=\sum_{i=0}^{m}\left\{{m \atop i}\right\}j!G_{n,i}$$

对于每次询问枚举i就好了。

考虑如何求g，我们可以发现有$g=A-gB$的形式，那么可以用$(B+1)^{-1}A$来求出g，这里要用到多项式求逆，当然你也可以继续使用分治FFT。

时间复杂度：$O(mn\log n+mT)$

3 fantasy

3.1 题目大意

一个字符串，尾部插入或删除字符，询问一个区间出现了多少次给定字符串。

3.2 100%算法

我们将串倒过来，那么就是在前端加入删除字符，很容易想到后缀平衡树，而询问一个区间就是询问两次前缀，对于后缀平衡树上的一个节点我们用一个vector存下子树中所有后缀的位置，询问的时候二分一下就好了。这个vector也很好维护，注意到后缀平衡树是重量平衡树，在重构标号的同时用归并重构vector就行了。

时间复杂度：插入是$O(\log^2 n)$的，询问是$O(|S|\log n + \log^2 n)$的，总复杂度$O(q\log^2 n + len\log n)$

考虑分别优化每一部分的复杂度，重量平衡树我们选择treap，重构vector的时候就只要重构旋转的那一条链就可以了，插入就可以降到$O(\log n)$。

我们可以在后缀平衡树上记录一些Lcp信息来使得每一层不需要暴力比较大小，因为实现比较复杂而难以描述，但较为容易思考这里略去具体方法，可以降到$O(|S|)$。

而二分也可以去掉，我们可以记录上一层的vector里的每一个元素对应子树的哪一个元素，这样就可以通过根的位置一层一层定位，就不需要二分了，可以降到$O(\log n)$。

时间复杂度：$O(q\log n + len)$，因为常数较大并不能跑出很好的效率。