NOIP2019 模拟赛 Solution

EndSaH

2019 年 8 月 9 日

1 最大流

人话版题意：n 点 m 边无向图，每点度数至多为 3，边容量为 1，求任意两点间最大流之和。

由于最大流等于最小割，且度数最多为 3，所以任意两点间最大流最大也只有 3。

考虑最大流为 1, 2, 3 时分别是什么情况。

1: 两点联通。

2: 两点在一个边双内。

3: 任意删去一条边，两点仍在一个边双内。

1, 2 随便搞，3 只需要枚举删哪条边然后 hash 其所在边双编号即可。

表面 $O(n^2 + m(n+m))$，实际常数很小。

2 拼数

考虑求有多少对 (i, j) 满足拼数结果为 k 的倍数。

可以发现，对 x, y 拼数的本质是 $x * 10^{\lfloor \log_{10} y \rfloor + 1} + y$（这个式子不严谨，明白意思就行）。

只要加号左边的与加号右边的对 k 取余的数能够加起来得到 k 或 0 即可。

考虑枚举 j，设 j 的位数为 $dig = \lfloor \log_{10} a_j \rfloor + 1$，那么只要查询有多少个数满足乘 10^{dig} 的结果对 k 取余为 $k - a_j \bmod k$（若 $k|a_j$，则为 0）。

由于值域限制，dig 最大只能到 10。于是开 10 个大小为 k 的桶，第 p 个桶的第 q 个位置表示有多少个数满足乘上 10^p 对 k 取余结果为 p。这个桶预处理很简单，枚举每个数乘 $10^{1...10}$ 算就可以了。k 大小比较大开 std::map 即可。

从 1 到 n 枚举 j 计算答案，注意计算前在桶内消去 j 的贡献。最后答案即 $n(n-1) - ans$，ans 为满足拼数结果为 k 的对数。

$O(n \log n \log SIZE)$，其中 $SIZE$ 表示值域大小。

另外记得 $10^9 * 10^{10}$ 会炸 long long，需要边乘边模 k。

3 火花

不知道有没有原题...

3.1 $fa_i = i - 1$

链状的在线，可以考虑直接倍增求答案

具体而言，处理出每个点往右走 2^i 步所经边权的最大值，然后询问的时候倍增往右跳，跳到一个最远的路径边权全部小于等于询问权值的点，于是前缀和相减即可

3.2 $n \le 10^5$

听别人口胡了一个可持久化平衡树的做法，~~没怎么听懂~~让我意识到这题可能会有一些 $\log^2$ 的解法，所以出了这一档。

3.3 $T = 0$

离线的话，考虑将询问和边按权值排序，每次将小于等于当前询问权值的边加入树中，然后用 LCT 做即可，每次就是查询一个点的子树和。

实现上可以化边权为点权，然后以子树和减去当前点点权作为答案。

但是这个做法并不好扩展到在线（你要是能码个可持久化 LCT 也行）。

所以考虑另一种做法，依然基于不断往树上加边的思想。

3.4 $T = 1$

如何在一个不完全的树上正确的统计子树和?

每次新加入一条边 u, fa，那么便将 fa 到**当前不完全的树**上能追溯到的最早祖先的子树和全部加上当前 u 的子树和。

~~这个正确性应该十分显然~~

最早祖先可以用并查集求，只有路径加，单点查便考虑树上差分，于是稍微应用一下 dfn 序就变成单点加和区间查。

然后你会惊奇的发现这个东西可以在线了，可持久化线段树就可以了。

这里可能出现一个问题，单点加，区间查直接可持久化线段树，但并查集呢? 因为有固定的合并方向（儿子并往父亲），所以不能启发式合并，并查集也就无法可持久化了。

但是实际上我们根据询问的权值回溯到加第 i 条边前的状态后，我们并不需要知道此时树的形态（也就是并查集的形态），而只是需要知道线段树的形态。

所以可以直接在建可持久化线段树的过程中用一个普通的路径压缩并查集算。

每次询问就 upper_bound 到需要回溯的位置，在那个节点的线段树查区间和就行。

（注意这里树上差分单点加只要加两次）

建树加询问时间复杂度共 $O((n+q)\log n)$，空间因为可持久化也带了 $\log$。