《过去的集合》解题报告

厦门双十中学 汪文潇

1 试题来源

2016年国家集训队互测

2 试题大意

有n个物品，每个物品有一个权值，进行m次操作，操作分成两种：

1、将某两个集合合并

2、询问某一次操作后指定集合的权值的和，这里加法是另外定义的，满足交换律和结合律。

通过一个交互库进行测试。这个库定义了物品类，提供了以物品为关键字到其他类型的映射表，对元素类型变量的赋值次数与访问映射表的次数之和count作出一定限制。

最大规模的数据点n=100000,m=500000,要求counter不超过8e6，时间限制3s，空间512mb。

3 主要算法介绍

这一道题本身是一个经典问题，即要求一个部分可持久化的并查集，支持对当前状态进行修改，以及询问历史版本的信息。

3.1 Task1

数据点1：n=100,m=500,要求counter不超过1e9。

不多赘述，读清题意即可，用以统计读题人数。

3.2 Task2

数据点2 3：n=10000,m=50000,要求counter不超过1e9,只询问当前局面

这部分即传统的并查集问题，借此机会整理一下并查集的一些经典方法。

对于普通的并查集问题，我们有经典的维护森林法，即通过连通性来表示是否在同一个集合，这时我们有两个常用优化：

1、路径压缩，复杂度是均摊$O(\log n)$

2、按秩合并，复杂度是最坏$O(\log n)$

同时我们也知道，结合这两个优化，复杂度是均摊$O(\alpha(n))$的。

3.3 Task3

数据点4 5：n=100000,m=500000,要求counter不超过1e9 这部分对count的要求是很松的，只需要考虑时间和空间限制即可。

3.3.1 算法1

有人说，并查集和可持久化的并查集之间其实只差一个可持久化。

那么我们直接实现一个可持久化的数组，然后用它实现并查集就可以了。

根据可持久化数组和并查集部分的不同实现，时空复杂度有所不同。需要注意的是，没有使用按秩合并的只使用路径压缩的并查集单次询问最坏情况是$O(n)$的，在此情况下并不可靠，尽管在很多数据下可能表现良好。

3.3.2 算法2

我们考虑从维护森林的角度进行思考，可以发现，在并查集的过程中，并在一起的集合是不会分开的，对应到森林上也即没有删边。我们假设合并两个集合时，就在对应两棵树的根间连一条边，任选其中之一作为另一个的父节点。考虑记录每条边出现的时间，那么从任意一点到根的路径上的边出现时间都是单调增的，那么我们就可以在询问时使用二分或者倍增的方法向上找到所询问时刻的根。

将当时集合的信息记录在当时的根上是很容易的，我们要解决的就是如何利用单调的性质找当时的根。我们需要支持个一个无父节点的点增加一个父节点，以及快速找到一个点到根路径上离根最近的边权小于询问值的边。LCT显然是可以胜任的，另一个方法是使用倍增。

对于每个点x，每个i，记录向根走2^i步所到节点$f_{x,i}$，以及能够走到y的最早的时刻。每次合并操作，实际上是给某个$f_{x,0}$赋值。当某个$f_{x,i}$被确定后，其值便不会再被修改，同时，对于所有满足$f_{y,i} = x$的y，$f_{y,i+1}$也将被确定。对于每对(x, i)，用链表将满足$f_{y,i} = x$的y记录下来，当$f_{x,i}$确定后，就可以根据这个链表更新出所有新确定的$f_{x,i}$。

这个做法修改的复杂度是均摊$O(\log n)$，询问的复杂度是最坏$O(\log n)$。如果实现良好可以通过Task4。

3.4 Task4

数据点6 7：n=100000,m=500000,要求counter不超过2.5×1e7

仍然考虑维护森林法，注意到按秩合并的情况下，树高最坏不超过$O(\log n)$。结合上面提到的性质，直接暴力走到最后一个满足出现时间比所询问时间早的边即可。修改和询问的复杂度都是是最坏$O(\log n)$，可以通过这一部分。

3.5 Task5

数据点8 10：n=100000,m=500000,要求counter不超过8×1e6

注意到按秩合并的方法能够保证树高不超过$O(\log n)$，而倍增的方法复杂度实际上是$O(\log p)$的，结合这两个方法，就得到了一个修改复杂度均摊$O(\log \log n)$，询问复杂度最坏$O(\log \log n)$的算法。

3.6 Extra

这题使用交互测试的初衷是为了通过计算次数来尽量方便地比较算法复杂度优劣，但实际上这个问题并没有通过交互完全等价于部分可持久化并查集这个经典问题。事实上选手仍然有办法在不清楚交互库的具体情况时通过自行标号和辅助点的方式来避过大部分的交互库的计数，从而达到线性复杂度的counter。遗憾的是，这似乎很难避免。

4 分数估计

7到9人拿到满分。

0到2人拿到70分

0到2人拿到50分

所有人都拿到至少30分