树上问题II

任飞宇

长链剖分

定义 **重子节点** 表示其子节点中子树深度最大的子结点。如果有多个子树最大的子结点，取其一。如果没有子节点，就无重子节点。

定义 **轻子节点** 表示剩余的子结点。

从这个结点到重子节点的边为 **重边**。

到其他轻子节点的边为 **轻边**。

若干条首尾衔接的重边构成 **重链**。

1. Blood Cousins

有一个家族关系树，描述了 n $(1 \leq n \leq 1e5)$ 人的家庭关系，成员编号为 1 到 n 。

如果 a 是 b 的父亲，那么称 a 为 b 的 1 级祖先；如果 b 有一个 1 级祖先，a 是 b 的 1 级祖先的 $(k-1)$ 级祖先，那么称 a 为 b 的 k 级祖先。

家庭关系保证是一棵树，树中的每个人都只有一个父母，且自己不会是自己的祖先。

如果存在一个人 z ，是两个人 a 和 b 共同的 p 级祖先：那么称 a 和 b 为 p 级表亲。

m $(1 \leq m \leq 1e5)$ 次询问，每次询问给出一对整数 v 和 p ，求编号为 v 的人有多少个 p 级表亲。

1. Blood Cousins

由于询问是离线的，可以把询问挂在对应的点上。自底向上处理每个点上的询问。

对每个点u求一个数组$f[u][i]$，表示以u为根的子树内有多少深度为i的点。

$f[u]$可以通过合并每个儿子的数组得到，在合并时把其他儿子的数组加入深度最大的儿子的数组中。

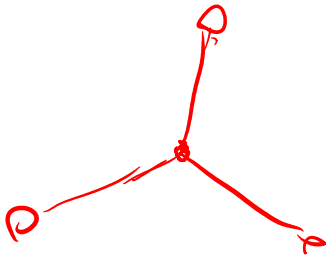
由于每次合并的复杂度等价于以轻子节点为顶端的长链的长度，而所有长链长度之和为$O(n)$，所以总复杂度是$O(n)$的。

具体实现可以使用vector。由于还需要加入当前点，即在数组开头插入一个数，我们可以在vector里倒序存数组。

重子节点的vector传给当前点不能直接复制，可以使用指针复制。

使用vector<int> f[n]来存vector，p[u]表示点u的vector所在的地址，即点u对应的vector是f[p[u]]。那么把v的vector传给u可以用p[u]=p[v]来实现。

2. Hotels



给定一棵树，在树上选3个点，要求两两距离相等，求方案数。

$n \leq 10^5$

2. Hotels

转为有根树，那么这3个点的形态一共3种。

在它们的顶点处统计数量，每次并入一颗子树就计算一部分在原先子树，一部分在被并入子树的方案数。

于是我们需要树形dp，$f_{i,j}$表示在i的子树中$d(x,i)=j$的x的数量，$g_{i,j}$表示在i的子树中，满足$d(lca(x,y),x)=d(lca(x,y),y)=d(lca(x,y),i)+j$的无序数对$(x,y)$的数量。

每个点的dp数组大小等于子树深度，且合并两个dp数组只需枚举较小的那个，所以可以使用和上题相同的优化，总复杂度$O(n)$。

线段树合并

3. Tree Rotations

给定一颗有 n 个**叶节点**的二叉树。每个叶节点都有一个权值 p_i（注意，根不是叶节点），所有叶节点的权值构成了一个 $1 \sim n$ 的排列。

对于这棵二叉树的任何一个结点，保证其要么是叶节点，要么左右两个孩子都存在。

现在你可以任选一些节点，交换这些节点的左右子树。

在最终的树上，按照先序遍历遍历整棵树并依次写下遇到的叶结点的权值构成一个长度为 n 的排列，你需要最小化这个排列的逆序对数。

3. Tree Rotations

给定一颗有 n 个**叶节点**的二叉树。每个叶节点都有一个权值 p_i（注意，根不是叶节点），所有叶节点的权值构成了一个 $1 \sim n$ 的排列。
对于这棵二叉树的任何一个结点，保证其要么是叶节点，要么左右两个孩子都存在。
现在你可以任选一些节点，交换这些节点的左右子树。
在最终的树上，按照先序遍历遍历整棵树并依次写下遇到的叶结点的权值构成一个长度为 n 的排列，你需要最小化这个排列的逆序对数。

考虑交换某个节点的两个儿子对最终答案的影响,交换之后对于子树内的逆序对并不会产生影响,对这颗子树树以外的逆序对也不会产生影响,所以产生的影响只有两颗子树.那么只需要考虑哪棵子树放前面更优就好了.

考虑用权值线段树来维护每个数出现的次数,考虑合并的时候同时计算出两种情况所产生的逆序对个数.

4. 树

给定一棵 n 个结点的有根树 T，结点从 1 开始编号，根结点为 1 号结点，每个结点有一个正整数权值 v_i。

设 x 号结点的子树内（包含 x 自身）的所有结点编号为 $c_1, c_2, \ldots, c_k$，定义 x 的价值为：

$$val(x) = (v_{c_1} + d(c_1, x)) \oplus (v_{c_2} + d(c_2, x)) \oplus \cdots \oplus (v_{c_k} + d(c_k, x))$$

其中 $d(x, y)$ 表示树上 x 号结点与 y 号结点间唯一简单路径所包含的边数，$d(x, x) = 0$。$\oplus$ 表示异或运算。

请你求出 $\sum_{i=1}^{n} val(i)$ 的结果。

4. 树

给定一棵 n 个结点的有根树 T，结点从 1 开始编号，根结点为 1 号结点，每个结点有一个正整数权值 v_i。

设 x 号结点的子树内（包含 x 自身）的所有结点编号为 $c_1, c_2, \ldots, c_k$，定义 x 的价值为：

$$val(x) = (v_{c_1} + d(c_1, x)) \oplus (v_{c_2} + d(c_2, x)) \oplus \cdots \oplus (v_{c_k} + d(c_k, x))$$

其中 $d(x, y)$ 表示树上 x 号结点与 y 号结点间唯一简单路径所包含的边数，$d(x, x) = 0$。$\oplus$ 表示异或运算。

请你求出 $\sum_{i=1}^{n} val(i)$ 的结果。

使用Trie维护x子树内所有点的$v_i + d(i,x)$异或和，于是需要实现：

1. 插入一个数
2. 所有数+1
3. 合并两个Trie
4. 求所有数的异或和

对于合并操作，只需要合并两个Trie的公共节点，复杂度同线段树合并。

5. Minimax

小 C 有一棵 n 个结点的有根树，根是 1 号结点，且每个结点最多有两个子结点。

定义结点 x 的权值为：

1.若 x 没有子结点，那么它的权值会在输入里给出，**保证这类点中每个结点的权值互不相同。**

2.若 x 有子结点，那么它的权值有 p_x 的概率是它的子结点的权值的最大值，有 $1-p_x$ 的概率是它的子结点的权值的最小值。

现在小 C 想知道，假设 1 号结点的权值有 m 种可能性，**权值第 i 小**的可能性的权值是 V_i，它的概率为 $D_i(D_i>0)$，求：

$$\sum_{i=1}^{m} i \cdot V_i \cdot D_i^2$$

你需要输出答案对 998244353 取模的值。

5. Minimax

我们设 $f_{i,j}$ 为 i 节点出现 j 的概率

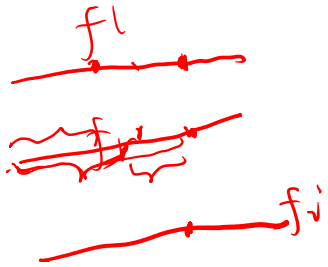
设 $l = ch[i][0], r = ch[i][1]$ 即左儿子右儿子

设 m 为叶子结点的个数

显然，i 出现 j 的概率为

$$f_{i,j} = f_{l,j} * (p_i \sum_{k=1}^{j-1} f_{r,k} + (1 - p_i) \sum_{k=j+1}^{m} f_{r,k}) + f_{r,j} * (p_i \sum_{k=1}^{j-1} f_{l,k} + (1 - p_i) \sum_{k=j+1}^{m} f_{l,k})$$

不难发现，这个柿子有关前缀和和后缀和，可以用线段树合并的操作来进行转移，从下到上转移，求出根节点的概率就好了



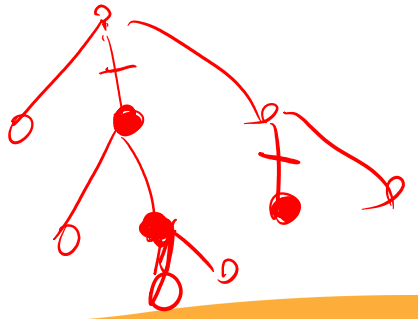
虚树

6. 消耗战

在一场战争中，战场由 n 个岛屿和 $n-1$ 个桥梁组成，保证每两个岛屿间有且仅有一条路径可达。现在，我军已经侦查到敌军的总部在编号为 1 的岛屿，而且他们已经没有足够多的能源维系战斗，我军胜利在望。已知在其他 k 个岛屿上有丰富能源，为了防止敌军获取能源，我军的任务是炸毁一些桥梁，使得敌军不能到达任何能源丰富的岛屿。由于不同桥梁的材质和结构不同，所以炸毁不同的桥梁有不同的代价，我军希望在满足目标的同时使得总代价最小。

侦查部门还发现，敌军有一台神秘机器。即使我军切断所有能源之后，他们也可以用那台机器。机器产生的效果不仅仅会修复所有我军炸毁的桥梁，而且会重新随机资源分布（但可以保证的是，资源不会分布到 1 号岛屿上）。不过侦查部门还发现了这台机器只能够使用 m 次，所以我们只需要把每次任务完成即可。

对于 100% 的数据，$2 \leq n \leq 2.5 \times 10^5, 1 \leq m \leq 5 \times 10^5, \sum k_i \leq 5 \times 10^5, 1 \leq k_i < n, h_i \neq 1, 1 \leq u, v \leq n, 1 \leq w \leq 10^5$ 。



6. 消耗战

考虑每个询问做一遍树形dp。
设 $f(i)$表示——使 i 不与其子树中任意一个关键点连通的最小代价。
枚举i的儿子v：

- 若v不是关键点：$f(i)=f(i)+\min\{f(v),w(i,v)\}$;
- 若v是关键点：$f(i)=f(i)+w(i,v)$;

时间复杂度$O(qn)$。

6. 消耗战

我们可以抽出关键点和它们的LCA，形成一颗更小的**虚树**，在虚树上进行dp。

用栈来维护当前虚树右侧的一条链上的点。

首先将根节点加入个栈，然后按DFS序从小到大添加关键点。

每次加入一个关键点，求出当前关键点与栈顶节点的LCA，把位于LCA下面的点全部弹栈，然后加入LCA和当前关键点即可。

```
rep(i,1,n){
        int t=a[i],f=lca(t,s[top]);;
        while (top){
            if (top>1&&dep[f]<dep[s[top-1]]) insert(s[top-1],s[top],dis(s[top-1],s[top])),top--;
            else if (dep[f]<dep[s[top]]) {insert(f,s[top],dis(f,s[top])); top--;break;}
            else break;
        }
        if (f!=s[top]) s[++top]=f;
        s[++top]=t;
    }
    while (top>1) insert(s[top-1],s[top],dis(s[top-1],s[top])),top--;
```

树的重心

定义：
对于树上的每一个点，计算其所有子树中最大的子树节点数，这个值最小的点就是这棵树的重心。（这里以及下文中的“子树”都是指无根树的子树，即包括“向上”的那棵子树，并且不包括整棵树自身。）

性质：
1. 以树的重心为根时，所有子树的大小都不超过整棵树大小的一半。
2. 如果有两个重心，它们一定相邻，可以分成大小相等的两颗子树；如果只有一个重心，所有子树的大小都严格小于整棵树大小的一半。
3. 树中所有点到某个点的距离和中，到重心的距离和是最小的；如果有两个重心，那么到它们的距离和一样。
4. 把两棵树通过一条边相连得到一棵新的树，那么新的树的重心在连接原来两棵树的重心的路径上。
5. 在一棵树上添加或删除一个叶子，那么它的重心最多只移动一条边的距离。

7. Kay and Snowflake

给定一颗n个点的有根树，求出以每个点为根的子树的重心。 $n \leq 3e5$

7. Kay and Snowflake

给定一颗n个点的有根树，求出以每个点为根的子树的重心。 $n \leq 3e5$

自底向上求每个子树的重心，设当前求点u的重心。

可以用反证法证明重心一定不在轻子树里。

重心一定在重子树的重心到点u的路径上。

暴力从重子树的重心往上枚举到u，判断是否是当前子树的重心即可，均摊复杂度$O(n)$。

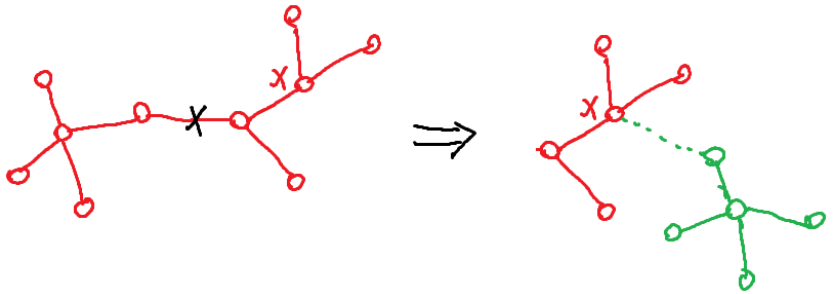
8. Centroids

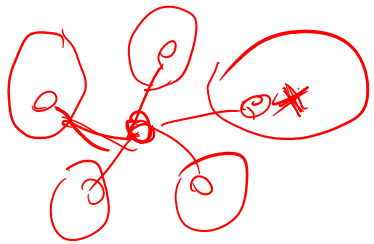
给定一颗树，你有一次将树改造的机会，改造的意思是删去一条边，再加入一条边，保证改造后还是一棵树。

请问有多少点可以通过改造，成为这颗树的重心？（如果以某个点为根，每个子树的大小都不大于$\frac{n}{2}$，则称某个点为重心）

8. Centroids

考虑原树有两个重心的情况，某个点x能否成为重心？ 一定可以

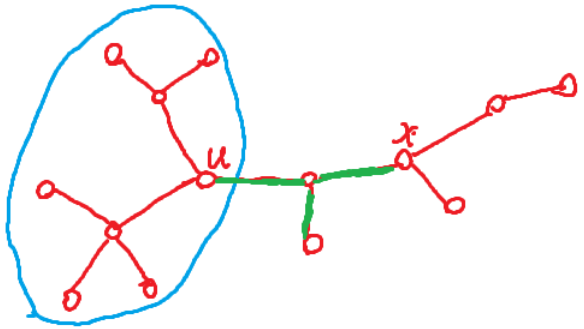




设原树只有一个重心u。点x不是重心，因为u所在的子树大小过半了，想让点x成为重心，就要切掉u所在子树中的一部分，挂到点x上。

显然切掉的部分在不超过$\frac{n}{2}$的条件下越大越好。

由于图中蓝色区域严格大于$\frac{n}{2}$，所以一定要切这里面的边，不能选绿色边。又因为切掉的部分在不超过$\frac{n}{2}$的条件下越大越好，所以一定切与u相邻的边里，切掉部分最大的。



9. Shaass the Great

树中有 n 个点，从 $n-1$ 条边中去除一条边，再构建一条相同长度的边重新构成一棵树（去除的边和构造的边可能相同），问新树中任意两点之间距离的总和最小是多少。

$n \leq 5000$。

9. Shaass the Great

树中有 n 个点，从 $n-1$ 条边中去除一条边，再构建一条相同长度的边重新构成一棵树（去除的边和构造的边可能相同），问新树中任意两点之间距离的总和最小是多少。

$n \leq 5000$。

枚举删哪条边，分成两颗树。

连上一条边(u,v)后，任意两点距离之和可以分为：

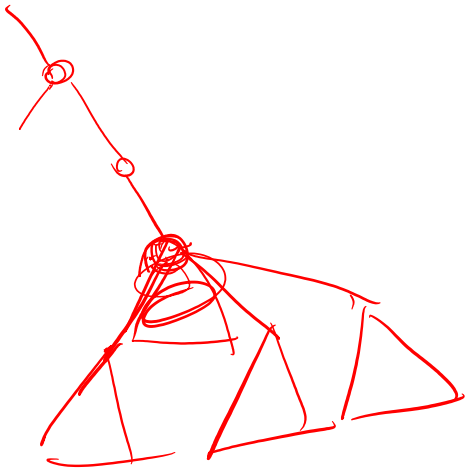
1. 两树内部距离和
2. $size_1 * size_2 * len_{u,v}$
3. 第一棵树所有点到u的距离和+第二棵树所有点到v的距离和

前两部分无论连哪两个点都是不变的，所以u，v应该选两颗树的重心

树分治

对于一些问题（尤其是路径相关问题），我们可以先处理重心相关的部分，然后递归子树，因为每颗子树的大小不超过整棵树大小的一半，所以递归的深度不超过$\log n$。

10. Ehab and the Big Finale



这是一道交互题。

你有一棵n个节点的有根树，1号点是根节点。

这棵树中有一个隐藏的节点x，你需要通过询问把x找出来。

你可以进行如下两种询问：

1、$d\ u\ (1 \leq u \leq n)$。交互库会返回节点u和x的距离。

2、$s\ u\ (1 \leq u \leq n)$。交互库会返回从u到x的路径上第二个点的标号。注意，你询问的u必须是x的祖先，否则会Wrong Ans。

你需要在不超过36次询问之内找出x。x是预先设定好的，不会随着询问而改变。

$2 \leq n \leq 2 \cdot 10^5$

10. Ehab and the Big Finale

尝试使用点分治。

求出当前范围内的重心c，判断x在哪颗子树内，继续递归下去。

要想用询问2来判断x在哪颗子树内，需要满足条件，即1号点和x不在同一子树内。

使用询问1获取$dis(c,x)$，若$dis(1,c)+dis(c,x)=dis(1,x)$，说明1号点和$x$不在同一子树内。

11. 成都七中

给你一棵 n 个节点的树，每个节点有一种颜色，有 m 次查询操作。

查询操作给定参数 l r x，需输出：

将树中编号在 $[l, r]$ 内的所有节点保留，x 所在连通块中颜色种类数。

每次查询操作独立。

11. 成都七中

对某个询问，考虑分治求解. 如果当前的分治中心和点x在同一个连通块，那么x所在的连通块就是分治中心所在的连通块，就在当前这一层处理这个询问. 之后就不用考虑这个询问了.

记每个点到分治中心路径上数字的max和min，就可以快速判断分治中心和点x是否在同一个连通块

每个询问等价于查询：当前分治区域中有哪些点满足到重心之间的路径上所有点编号在[l,r]内，求这些点的颜色数. 转为二维数点问题

扫描线扫一维，另一维需要单点插入，求前缀颜色数。我们只在每种颜色的第一个位置放一个1，那就是单点加减，求前缀和。
总复杂度$O(nlog^2n+qlogn)$