树相关杂题选讲

hk-cnyali

雅礼中学

2018 年 10 月 19 日

写在前面

今天主要讲一些树上问题的简单题目

写在前面

今天主要讲一些树上问题的简单题目

由于本人比较菜,能力有限,还是第一次讲题,有讲得不好的地方还请大家多多指正多多包涵

写在前面

今天主要讲一些树上问题的简单题目

由于本人比较菜,能力有限,还是第一次讲题,有讲得不好的地方还请大家多多指正多多包涵

题目很水,并没有NOIp超纲知识点,大佬们就当切切水题放松放松好了

Description

给出两棵n个节点的树，每次操作可以把第一棵树的一条边删掉然后连另外两个点，且必须满足每次操作完之后仍是一棵树

问最少需要多少步操作才能把第一棵树变成第二棵树(是完全相同,并不是同构),并输出方案

$n \leq 500000$

Solution

考虑两棵树都有的边，显然这些边对答案是没有影响的

Solution

考虑两棵树都有的边，显然这些边对答案是没有影响的

考虑把这些边两端的点用并查集都缩起来,这样会形成两棵由很多个新点形成的树

Solution

考虑两棵树都有的边，显然这些边对答案是没有影响的

考虑把这些边两端的点用并查集都缩起来,这样会形成两棵由很多个新点形成的树

每次从叶子节点开始自底向上操作,把第一棵树中这个新点连向父亲的边删去,在第二棵树中找到其对应的父亲,连起来

这样做可以保证不出现环,$O(n\alpha)$

Description

给出一棵n个节点的树和树上的m条路径.选出一个节点集合S,使得每条路径上至少有一个点在集合中,并使集合大小最小.

$n, m \leq 10^5$

Solution

可以发现一定存在一种最优方案只选择某些路径的lca.

因为如果存在一个S中的节点x不为任何一对节点的lca,那么经过x的路径总会经过x的父亲,所以我们可以选择x的父亲.以此类推,因此选择的节点必定是某些lca.

Solution

可以发现一定存在一种最优方案只选择某些路径的lca.

因为如果存在一个S中的节点x不为任何一对节点的lca,那么经过x的路径总会经过x的父亲,所以我们可以选择x的父亲.以此类推,因此选择的节点必定是某些lca.

所以直接贪心:按照路径lca的深度从大到小考虑,对于某一条路径,看在已经选出的集合中存不存在一个点在这条路径上,如果没有则把lca加到集合中.用树链剖分维护即可

Description

给出一棵n个节点的带边权的树，找出一条长度不超过k的链，最小化其他所有点到这条链的最短距离的最大值,求这个最小值

$k \leq n \leq 10^5$

Solution

结论:选出的k个点一定在直径上

Solution

结论:选出的k个点一定在直径上

证明:首先,直径最中间那个点是必须要选的,否则答案会大于直径的一半.考虑接下来选择的一个点,它一定是相邻的直径点.如果不是的话,由于选择的必须是一条链,所以接下来都不能往那个直径点方向扩张,但显然往直径点方向的深度更深,答案不会更优.

Solution

结论:选出的k个点一定在直径上

证明:首先,直径最中间那个点是必须要选的,否则答案会大于直径的一半.考虑接下来选择的一个点,它一定是相邻的直径点.如果不是的话,由于选择的必须是一条链,所以接下来都不能往那个直径点方向扩张,但显然往直径点方向的深度更深,答案不会更优.

接下来就能随便做了,可以直接二分答案,也可以用单调队列维护滑动窗口,维护直径两个端点到选出的k个点的端点的距离以及以这k个点为根的子树的最大深度的最大值即可,$O(n)$

Description

给定一颗n个点的树，i号点的权值是2^i

要求删去k个点，使得剩下的点仍然连通，并且总权值和最大

$k < n \leq 10^6$

Solution

可以将题意转化为保留$n-k$个点,使得保留点的总权值和最大.

Solution

可以将题意转化为保留$n-k$个点,使得保留点的总权值和最大.

因为$2^n > \sum_{i=0}^{n-1} 2^i$,显然需要将权值从大到小依次贪心考虑当前点是否保留,并把保留的点做标记

Solution

可以将题意转化为保留$n-k$个点,使得保留点的总权值和最大.

因为$2^n > \sum_{i=0}^{n-1} 2^i$,显然需要将权值从大到小依次贪心考虑当前点是否保留,并把保留的点做标记

首先n号点肯定要选,把n号点作为树的根,从大到小考虑每一个点i.

如果要选i的话,那么它到根路径上所有点都要选,所以只需要判断它到最近的已经被选的祖先的距离与剩下还能保留的点数的大小关系就能判断这个点是否能选了

Solution

可以将题意转化为保留$n-k$个点,使得保留点的总权值和最大.

因为$2^n > \sum_{i=0}^{n-1} 2^i$,显然需要将权值从大到小依次贪心考虑当前点是否保留,并把保留的点做标记

首先n号点肯定要选,把n号点作为树的根,从大到小考虑每一个点i.

如果要选i的话,那么它到根路径上所有点都要选,所以只需要判断它到最近的已经被选的祖先的距离与剩下还能保留的点数的大小关系就能判断这个点是否能选了

每次暴力打标记,倍增预处理祖先即可,$O(n \log n)$

Description

一开始有一个节点1,它的权值为v_1,有q次操作:

- $1\ p\ v$ 添加一个新的节点,其父亲为p,权值为v
- $2\ u$ 查询u的得分

  定义u的得分为:

  $ans_u = (\sum_{i|\text{i is u's son}} ans_i + v_u) * (size_u + 1)$

$q \leq 2 * 10^5$

Solution

首先考虑每个点对根节点的贡献c_i, 它显然是v_i的整数倍,即可以表示成$c_i = v_i * k_i$这样的形式

这其中的$k_i = \prod_{f|\text{f is i's anc}}(size_f + 1)$,即它到根所有节点的$(size + 1)$的乘积

Solution

首先考虑每个点对根节点的贡献c_i, 它显然是v_i的整数倍,即可以表示成$c_i = v_i * k_i$这样的形式

这其中的$k_i = \prod_{f|\text{f is i's anc}}(size_f + 1)$,即它到根所有节点的$(size + 1)$的乘积

那么它对某个祖先r的贡献即为$\frac{c_i}{k_{far}}$

Solution

首先考虑每个点对根节点的贡献c_i, 它显然是v_i的整数倍,即可以表示成$c_i = v_i * k_i$这样的形式

这其中的$k_i = \prod_{f|\text{f is i's anc}}(size_f + 1)$,即它到根所有节点的$(size + 1)$的乘积

那么它对某个祖先r的贡献即为$\frac{c_i}{k_{far}}$

先把树的形态存下来,一开始每个点权值为0,再对权值进行更新.

Solution

首先考虑每个点对根节点的贡献c_i, 它显然是v_i的整数倍,即可以表示成$c_i = v_i * k_i$这样的形式

这其中的$k_i = \prod_{f|\text{f is i's anc}}(size_f + 1)$,即它到根所有节点的$(size + 1)$的乘积

那么它对某个祖先r的贡献即为$\frac{c_i}{k_{far}}$

先把树的形态存下来,一开始每个点权值为0,再对权值进行更新.

如果修改点u的权值,首先它自己的c_i值要赋成$v_i * k_i$,并且它父亲f的子树中所有点的c_i和k_i都要乘以$\frac{size_f+2}{size_f+1}$

Solution

首先考虑每个点对根节点的贡献c_i, 它显然是v_i的整数倍,即可以表示成$c_i = v_i * k_i$这样的形式

这其中的$k_i = \prod_{f|\text{f is i's anc}}(size_f + 1)$,即它到根所有节点的$(size + 1)$的乘积

那么它对某个祖先r的贡献即为$\frac{c_i}{k_{fa_r}}$

先把树的形态存下来,一开始每个点权值为0,再对权值进行更新.

如果修改点u的权值,首先它自己的c_i值要赋成$v_i * k_i$,并且它父亲f的子树中所有点的c_i和k_i都要乘以$\frac{size_f+2}{size_f+1}$

查询u点的答案即为u子树中c_i和的除以k_{fa_u}

用线段树维护每个点的c_i和k_i值即可,$O(n \log n)$

Description

给出一棵n个节点的树,对于每个点i有q_i的概率直接带电,对于每条边(a,b), 有$p_{a,b}$的概率通电

求带电的节点期望个数

$n \leq 5 * 10^5$

Solution

带电的节点期望个数实际上就是每个点带电概率的和

Solution

带电的节点期望个数实际上就是每个点带电概率的和

考虑dp,每个点带电只有可能是它父亲带电或是它的儿子(或是自己)带电.

Solution

带电的节点期望个数实际上就是每个点带电概率的和

考虑dp,每个点带电只有可能是它父亲带电或是它的儿子(或是自己)带电.

由于不好直接计算带电的概率,设f_u为u不被它儿子(或是自己)充电的概率,g_u为它不被它父亲充电的概率

Solution

带电的节点期望个数实际上就是每个点带电概率的和

考虑dp,每个点带电只有可能是它父亲带电或是它的儿子(或是自己)带电.

由于不好直接计算带电的概率,设f_u为u不被它儿子(或是自己)充电的概率,g_u为它不被它父亲充电的概率

- 求f_u

Solution

带电的节点期望个数实际上就是每个点带电概率的和

考虑dp,每个点带电只有可能是它父亲带电或是它的儿子(或是自己)带电.

由于不好直接计算带电的概率,设f_u为u不被它儿子(或是自己)充电的概率,g_u为它不被它父亲充电的概率

- 求f_u

$$f_u = (1 - q_u) * \prod_{v|\text{v is u's son}} (f_v + (1 - f_v) * (1 - p_{u,v}))$$

也就是本身自己必须没电,并且子节点没电或是子节点带电但导线不导电

Solution

- 求g_u

Solution

- 求g_u

  设P表示当前v的父亲u不向它供电的概率,那么

$$P = \frac{f_u * g_u}{f_v + (1 - f_v) * (1 - p_{u,v})}$$

Solution

- 求g_u

  设P表示当前v的父亲u不向它供电的概率,那么

$$P = \frac{f_u * g_u}{f_v + (1 - f_v) * (1 - p_{u,v})}$$

  因为这时$f_u * g_u$是由v及v的子树推过来的,如果再推回去的话会被算重,所以要除掉一个从v及其子树推上来的概率

Solution

- 求g_u

  设P表示当前v的父亲u不向它供电的概率,那么

$$P = \frac{f_u * g_u}{f_v + (1 - f_v) * (1 - p_{u,v})}$$

  因为这时$f_u * g_u$是由v及v的子树推过来的,如果再推回去的话会被算重,所以要除掉一个从v及其子树推上来的概率

  于是就有

$$g_v = P + (1 - P) * (1 - p_{u,v})$$

Solution

- 求g_u

设P表示当前v的父亲u不向它供电的概率,那么

$$P = \frac{f_u * g_u}{f_v + (1 - f_v) * (1 - p_{u,v})}$$

因为这时$f_u * g_u$是由v及v的子树推过来的,如果再推回去的话会被算重,所以要除掉一个从v及其子树推上来的概率

于是就有

$$g_v = P + (1 - P) * (1 - p_{u,v})$$

答案显然是$\sum_{i=1}^{n}(1 - g_i * f_i)$

$O(n)$

Description

给定一棵n个节点的完全二叉树,每个点有权值a_i,边有权值b_i

你需要按某种顺序访问这个n个节点,新访问一个点的花费为上一个被访问的点到它的距离,乘以这个点的权值.

访问顺序必须满足:

- 在任意时刻,访问过的点必须是一个连通块
- 访问了某个点后必须接着把它的子树访问完

求访问所有节点的最小花费

$n \leq 2 * 10^5$

Solution

考虑一条合法的行走路径，必定是先到一个点，把它的子树走完，然后从子树中最后一个被访问的点走回到它的父亲，再走它的兄弟以及兄弟的子树,以此类推

Solution

考虑一条合法的行走路径，必定是先到一个点，把它的子树走完，然后从子树中最后一个被访问的点走回到它的父亲，再走它的兄弟以及兄弟的子树,以此类推

那么本质就是要求从某个点出发，访问完其子树后回到其某个祖先的最小代价

Solution

考虑一条合法的行走路径，必定是先到一个点，把它的子树走完，然后从子树中最后一个被访问的点走回到它的父亲，再走它的兄弟以及兄弟的子树,以此类推

那么本质就是要求从某个点出发，访问完其子树后回到其某个祖先的最小代价

由于是完全二叉树,深度有保证,可以直接设$g_{x,i}$表示从x开始,访问完x的子树后再走到深度为i的祖先(设为u)的最小代价

Solution

- x为叶子节点: $g_{x,i} = dist(x, u) * a_u$

Solution

- x为叶子节点: $g_{x,i} = dist(x,u) * a_u$
- x只有左儿子: $g_{x,i} = g_{lson,i} + dist(x,lson) * a_{lson}$

Solution

- x为叶子节点: $g_{x,i} = dist(x,u) * a_u$
- x只有左儿子: $g_{x,i} = g_{lson,i} + dist(x,lson) * a_{lson}$
- x有两个儿子: 枚举先访问左子树还是右子树

Solution

- x为叶子节点: $g_{x,i} = dist(x,u) * a_u$
- x只有左儿子: $g_{x,i} = g_{lson,i} + dist(x,lson) * a_{lson}$
- x有两个儿子: 枚举先访问左子树还是右子树

但是最后一种情况还是不好直接计算,再设一个$f_{x,i}$表示从x开始,访问完x的子树后再走到深度为i的祖先的另外一个儿子的最小代价

Solution

- x为叶子节点: $g_{x,i} = dist(x,u) * a_u$
- x只有左儿子: $g_{x,i} = g_{lson,i} + dist(x,lson) * a_{lson}$
- x有两个儿子: 枚举先访问左子树还是右子树

但是最后一种情况还是不好直接计算,再设一个$f_{x,i}$表示从x开始,访问完x的子树后再走到深度为i的祖先的另外一个儿子的最小代价

f的转移和g类似,最后再用f去转移g

Solution

- x为叶子节点: $g_{x,i} = dist(x,u) * a_u$
- x只有左儿子: $g_{x,i} = g_{lson,i} + dist(x,lson) * a_{lson}$
- x有两个儿子: 枚举先访问左子树还是右子树

但是最后一种情况还是不好直接计算,再设一个$f_{x,i}$表示从x开始,访问完x的子树后再走到深度为i的祖先的另外一个儿子的最小代价

f的转移和g类似,最后再用f去转移g

枚举第一个访问的点,模拟访问过程统计答案即可,$O(n \log n)$

Description

给定一棵n个节点的树.对于树上每个结点,将它删去,然后你可以将得到的森林中任意一棵树的一个点与其父亲断开并连接到另一颗树上.

最小化森林中所有树$size$最大值

$n \leq 10^5$

Solution

考虑求删掉某个点时的答案,显然可以二分答案.

Solution

考虑求删掉某个点时的答案,显然可以二分答案.

记max为此时森林中最大树的大小,min同理

那么对于二分的一个答案ans,我们就是要判断在最大的树中是否存在一个节点i,满足$size_i + min \leq ans$且$max - size_i \leq ans$.

即$max - ans \leq size_i \leq ans - min$

Solution

考虑求删掉某个点时的答案,显然可以二分答案.

记max为此时森林中最大树的大小,min同理

那么对于二分的一个答案ans,我们就是要判断在最大的树中是否存在一个节点i,满足$size_i + min \leq ans$且$max - size_i \leq ans$.

即$max - ans \leq size_i \leq ans - min$

很容易想到用map维护每个点的$size$来判断

Solution

考虑求删掉某个点时的答案,显然可以二分答案.

记max为此时森林中最大树的大小,min同理

那么对于二分的一个答案ans,我们就是要判断在最大的树中是否存在一个节点i,满足$size_i + min \leq ans$且$max - size_i \leq ans$.

即$max - ans \leq size_i \leq ans - min$

很容易想到用map维护每个点的$size$来判断

维护4个map:

- map_{anc}: i所有祖先的$size$
- map_{heavy}: i重儿子($size$最大的儿子)子树中所有点的$size$
- map_{light}: i轻儿子子树中所有点的$size$
- map_{other}: 除了上述情况剩下所有点的$size$

Solution

求i点的答案时,判断删掉它之后的森林中最大的树位于哪里:

Solution

求i点的答案时,判断删掉它之后的森林中最大的树位于哪里:

- i点的重儿子: 在map_{heavy}上查询

Solution

求i点的答案时,判断删掉它之后的森林中最大的树位于哪里:

- i点的重儿子: 在map_{heavy}上查询
- 根节点所在树: 在map_{anc}和map_{other}上查询,其中map_{anc}中的值要减去$size_i$

Solution

求i点的答案时,判断删掉它之后的森林中最大的树位于哪里:

- i点的重儿子: 在map_{heavy}上查询
- 根节点所在树: 在map_{anc}和map_{other}上查询,其中map_{anc}中的值要减去$size_i$

如何维护这4个map:

- map_{anc}: dfs时更新,退出时删除
- map_{heavy}, map_{light}: 启发式合并
- map_{other}: 一开始把所有点加入,其他map加入一个点时就删除这个点

$O(n \log^2 n)$

Description

给定一棵n个节点的树以及一个长为m的序列，序列每个位置上的值$\in [1,n]$

定义序列中一个长度为偶数的区间的代价为这个区间所有数在树上两两匹配,匹配的节点两两距离和的最小值

你需要求出序列中所有长度为偶数的区间代价之和对998244353取模的结果

$n,m \leq 10^5$

Solution

首先对于一个确定的区间,我们显然会让其中的点在尽可能深的位置匹配.也就是说,每颗子树中未匹配的点最多只有一个

Solution

首先对于一个确定的区间,我们显然会让其中的点在尽可能深的位置匹配.也就是说,每颗子树中未匹配的点最多只有一个

考虑对于一条x的父亲边,它对最后的答案会被计算贡献当且仅当x的子树里有奇数个选出的点.题目就可以转化为对于每一颗子树,有多少个长度为偶数的区间使得子树中有奇数个区间内的点

Solution

首先对于一个确定的区间,我们显然会让其中的点在尽可能深的位置匹配.也就是说,每颗子树中未匹配的点最多只有一个

考虑对于一条x的父亲边,它对最后的答案会被计算贡献当且仅当x的子树里有奇数个选出的点.题目就可以转化为对于每一颗子树,有多少个长度为偶数的区间使得子树中有奇数个区间内的点

把这棵子树中的点在原序列中记为1,不在子树中的记为0,并求一遍前缀和,那么要求的东西就是满足$s_r - s_{l-1} \equiv 1 \pmod 2$且$l - 1 \equiv r \pmod 2$的$(l, r)$的对数

Solution

首先对于一个确定的区间,我们显然会让其中的点在尽可能深的位置匹配.也就是说,每颗子树中未匹配的点最多只有一个

考虑对于一条x的父亲边,它对最后的答案会被计算贡献当且仅当x的子树里有奇数个选出的点.题目就可以转化为对于每一颗子树,有多少个长度为偶数的区间使得子树中有奇数个区间内的点

把这棵子树中的点在原序列中记为1,不在子树中的记为0,并求一遍前缀和,那么要求的东西就是满足$s_r - s_{l-1} \equiv 1 \pmod 2$且$l - 1 \equiv r \pmod 2$的$(l, r)$的对数

用线段树维护在偶数/奇数位置上前缀和为0/1的个数.每加进来一个数就相当于对一个后缀区间反转,那么线段树$pushup$时通过左半边区间原始序列中1的个数的奇偶来决定是否对右区间反转

Solution

首先对于一个确定的区间,我们显然会让其中的点在尽可能深的位置匹配.也就是说,每颗子树中未匹配的点最多只有一个

考虑对于一条x的父亲边,它对最后的答案会被计算贡献当且仅当x的子树里有奇数个选出的点.题目就可以转化为对于每一颗子树,有多少个长度为偶数的区间使得子树中有奇数个区间内的点

把这棵子树中的点在原序列中记为1,不在子树中的记为0,并求一遍前缀和,那么要求的东西就是满足$s_r - s_{l-1} \equiv 1 \pmod 2$且$l - 1 \equiv r \pmod 2$的$(l, r)$的对数

用线段树维护在偶数/奇数位置上前缀和为0/1的个数.每加进来一个数就相当于对一个后缀区间反转,那么线段树$pushup$时通过左半边区间原始序列中1的个数的奇偶来决定是否对右区间反转

对每个子树都要维护，直接线段树合并即可,$O(n \log n)$

Thanks