杂题选讲

by zyb

2019 年 6 月 2 日

problem A

- 给一棵n个点的树

problem A

- 给一棵n个点的树
- 每个点都有一个点权a_i, a_i互不相同

problem A

- 给一棵n个点的树
- 每个点都有一个点权a_i, a_i互不相同
- 定义树上的两个点(x, y)，如果x是y的祖先，且$a_x > a_y$，则称(x, y)是一对逆序对

problem A

- 给一棵n个点的树
- 每个点都有一个点权a_i, a_i互不相同
- 定义树上的两个点(x, y)，如果x是y的祖先，且$a_x > a_y$，则称(x, y)是一对逆序对
- 你可以指定若干个点，将它的点权取反

problem A

- 给一棵n个点的树
- 每个点都有一个点权a_i, a_i互不相同
- 定义树上的两个点(x, y)，如果x是y的祖先，且$a_x > a_y$，则称(x, y)是一对逆序对
- 你可以指定若干个点，将它的点权取反
- 对于$1 - m$里的每个数k，问是否有一种方案使得逆序对数恰好为k

problem A

- 给一棵n个点的树
- 每个点都有一个点权a_i, a_i互不相同
- 定义树上的两个点(x, y)，如果x是y的祖先，且$a_x > a_y$，则称(x, y)是一对逆序对
- 你可以指定若干个点，将它的点权取反
- 对于$1 - m$里的每个数k，问是否有一种方案使得逆序对数恰好为k
- $1 \leq n, m \leq 10^5$

problem A

- 对于一组有祖先关系的点对(x, y)，他们是否是逆序对只取决于a的绝对值较大的那个点

problem A

- 对于一组有祖先关系的点对(x, y)，他们是否是逆序对只取决于a的绝对值较大的那个点
- 所以只需考虑每一个点决定的点对数量即可

problem A

- 对于一组有祖先关系的点对(x, y)，他们是否是逆序对只取决于a的绝对值较大的那个点
- 所以只需考虑每一个点决定的点对数量即可
- 即每个点子树内有多少个点的权值比它小，以及它到根路径上有多少个点的权值比它小

problem A

- 对于一组有祖先关系的点对(x, y)，他们是否是逆序对只取决于a的绝对值较大的那个点
- 所以只需考虑每一个点决定的点对数量即可
- 即每个点子树内有多少个点的权值比它小，以及它到根路径上有多少个点的权值比它小
- 这两个问题都是经典问题，可以在$O(n \log n)$的时间内求出

problem A

- 对于点x，设它取反和不取反的贡献分别是A, B

problem A

- 对于点x，设它取反和不取反的贡献分别是A, B
- 假设$A \leq B$，可以先把A加到答案中去，然后对一堆$B - A$的物品做01背包

problem A

- 对于点x，设它取反和不取反的贡献分别是A, B
- 假设$A \le B$，可以先把A加到答案中去，然后对一堆$B - A$的物品做01背包
- 用*bitset*加速

problem A

- 对于点x，设它取反和不取反的贡献分别是A, B
- 假设$A \leq B$，可以先把A加到答案中去，然后对一堆$B - A$的物品做01背包
- 用$bitset$加速
- 时间复杂度$O(n\log n + n\frac{m}{64})$

problem B

- 给一个n个点带点权的树

problem B

- 给一个n个点带点权的树
- 问有多少个点集，满足点权和恰好为k，且任意两个点在树上不相邻

problem B

- 给一个n个点带点权的树
- 问有多少个点集，满足点权和恰好为k，且任意两个点在树上不相邻
- $1 \leq n \leq 50, 1 \leq k \leq 5000$

problem B

- 先对树轻重链剖分，然后进行背包

problem B

- 先对树轻重链剖分，然后进行背包
- 背包的顺序是从根开始*dfs*，先*dfs*轻儿子，再*dfs*重儿子

problem B

- 先对树轻重链剖分，然后进行背包
- 背包的顺序是从根开始*dfs*，先*dfs*轻儿子，再*dfs*重儿子
- 背包的时候还要记一下重链顶端的父亲节点是否被选取

problem B

- 先对树轻重链剖分，然后进行背包
- 背包的顺序是从根开始*dfs*，先*dfs*轻儿子，再*dfs*重儿子
- 背包的时候还要记一下重链顶端的父亲节点是否被选取
- 因为重链中间的点是否被选取对后面没有影响

problem B

- 先对树轻重链剖分，然后进行背包
- 背包的顺序是从根开始*dfs*，先*dfs*轻儿子，再*dfs*重儿子
- 背包的时候还要记一下重链顶端的父亲节点是否被选取
- 因为重链中间的点是否被选取对后面没有影响
- 因为每个点到根路径上最多只有$\log n$条重链，故背包的状态数为$O(2^{\log n} * m)$

problem B

- 先对树轻重链剖分，然后进行背包
- 背包的顺序是从根开始*dfs*，先*dfs*轻儿子，再*dfs*重儿子
- 背包的时候还要记一下重链顶端的父亲节点是否被选取
- 因为重链中间的点是否被选取对后面没有影响
- 因为每个点到根路径上最多只有$\log n$条重链，故背包的状态数为$O(2^{\log n} * m)$
- 总时间复杂度为$O(n^2 m)$

1. problem A
   - problem A
2. problem B
   - problem B
3. problem C
   - problem C
4. problem D
   - problem D
5. problem E
   - problem E
6. problem F
   - problem F

problem C

- 给一个长度为n的排列p

problem C

- 给一个长度为n的排列p
- 初始时$p_i = i$

problem C

- 给一个长度为n的排列p
- 初始时$p_i = i$
- 共Q次修改，每次修改会给$l, r, k (k \le r - l + 1)$，将$[l, r]$中的前$k$个数移动到这个区间的末尾
  即将$p_l, p_{l+1}, \dots, p_r$变成$p_{k+1}, p_{k+2}, \dots, p_r, p_l, p_{l+1}, \dots, p_k$

problem C

- 给一个长度为n的排列p
- 初始时$p_i = i$
- 共Q次修改，每次修改会给$l, r, k(k \leq r - l + 1)$，将$[l, r]$中的前$k$个数移动到这个区间的末尾
  即将$p_l, p_{l+1}, \ldots, p_r$变成$p_{k+1}, p_{k+2}, \ldots, p_r, p_l, p_{l+1}, \ldots, p_k$
- 每次修改完之后，询问将p的前多少个移动到末尾后，整个排列的逆序对个数最少

problem C

- 给一个长度为n的排列p
- 初始时$p_i = i$
- 共Q次修改，每次修改会给$l, r, k(k \leq r - l + 1)$，将$[l, r]$中的前$k$个数移动到这个区间的末尾
  即将$p_l, p_{l+1}, \ldots, p_r$变成$p_{k+1}, p_{k+2}, \ldots, p_r, p_l, p_{l+1}, \ldots, p_k$
- 每次修改完之后，询问将p的前多少个移动到末尾后，整个排列的逆序对个数最少
- $1 \leq n, Q \leq 10^5$

problem C

- 考虑将p的前k个移动到末尾后逆序对数的变化

problem C

- 考虑将p的前k个移动到末尾后逆序对数的变化
- 将p的前k个移动到末尾后逆序对数会减少$\sum_{i=1}^{k}(2*p_i-n-1)$

problem C

- 考虑将p的前k个移动到末尾后逆序对数的变化
- 将p的前k个移动到末尾后逆序对数会减少$\sum_{i=1}^{k}(2*p_i-n-1)$
- 把第i个点的点权设为$2*p_i-n-1)$，每次修改后相当于求最大前缀和的位置

problem C

- 考虑将p的前k个移动到末尾后逆序对数的变化
- 将p的前k个移动到末尾后逆序对数会减少$\sum_{i=1}^{k}(2*p_i-n-1)$
- 把第i个点的点权设为$2*p_i-n-1$)，每次修改后相当于求最大前缀和的位置
- 可以用平衡树维护

problem C

- 考虑将p的前k个移动到末尾后逆序对数的变化
- 将p的前k个移动到末尾后逆序对数会减少$\sum_{i=1}^{k}(2*p_i-n-1)$
- 把第i个点的点权设为$2*p_i-n-1)$，每次修改后相当于求最大前缀和的位置
- 可以用平衡树维护
- 时间复杂度$O(Q\log n)$

problem D

- 有一颗边无权的树。设树的大小为T,现在需要你找到一个1-T的排列P，使得 $\sum_{i=1}^{T}\frac{d(i,P(i))}{2}$ 最大。
  一开始树上只有1号点，现在支持动态向树中加点，保证加点始终是一棵树。其中的d(x,y)表示x和y在树上的距离。

  样例解释:
  第一次操作后树为1-2，选P=[2,1] d(1,2)/2+d(2,1)/2=0.5+0.5=1
  第二次操作后树为1-2-3，选P=[2,3,1]
  d(1,2)/2+d(2,3)/2+d(3,1)/2=0.5+0.5+1=2
  第三次操作后树为4-1-2-3，选P=[2,1,4,3]
  d(1,2)/2+d(2,1)/2+d(3,4)/2+d(4,3)/2=0.5+0.5+1.5+1.5=4

problem D

- 有一颗边无权的树。设树的大小为T,现在需要你找到一个1-T的排列P，使得 $\sum_{i=1}^{T} \frac{d(i,P(i))}{2}$ 最大。
一开始树上只有1号点，现在支持动态向树中加点，保证加点始终是一棵树。其中的d(x,y)表示x和y在树上的距离。

  样例解释:
  第一次操作后树为1-2，选P=[2,1] d(1,2)/2+d(2,1)/2=0.5+0.5=1
  第二次操作后树为1-2-3，选P=[2,3,1]
  d(1,2)/2+d(2,3)/2+d(3,1)/2=0.5+0.5+1=2
  第三次操作后树为4-1-2-3，选P=[2,1,4,3]
  d(1,2)/2+d(2,1)/2+d(3,4)/2+d(4,3)/2=0.5+0.5+1.5+1.5=4

- $1 \leq n \leq 10^5$

problem D

- 考虑每一条边的贡献，假设某一条边把这棵树分成大小为x, y的两个部分，则这条边的贡献就是$2 \times min(x, y)$

problem D

- 考虑每一条边的贡献，假设某一条边把这棵树分成大小为x, y的两个部分，则这条边的贡献就是$2 \times min(x, y)$
- 这个等于所有点到重心的距离和的两倍

problem D

- 考虑每一条边的贡献，假设某一条边把这棵树分成大小为x, y的两个部分，则这条边的贡献就是$2 \times min(x, y)$
- 这个等于所有点到重心的距离和的两倍
- 证明：

problem D

- 考虑每一条边的贡献，假设某一条边把这棵树分成大小为x, y的两个部分，则这条边的贡献就是$2 \times min(x, y)$
- 这个等于所有点到重心的距离和的两倍
- 证明：
- 以重心为根建树，假设一条边$a - b$, b是a的父亲，则这条边的贡献就是$2 \times a$的子树大小

problem D

- 考虑每一条边的贡献，假设某一条边把这棵树分成大小为x, y的两个部分，则这条边的贡献就是$2 \times min(x, y)$
- 这个等于所有点到重心的距离和的两倍
- 证明：
- 以重心为根建树，假设一条边$a - b$, b是a的父亲，则这条边的贡献就是$2 \times a$的子树大小
- =所有点深度和*2 =所有点到重心的距离和*2

problem D

- 考虑每一条边的贡献，假设某一条边把这棵树分成大小为x, y的两个部分，则这条边的贡献就是$2 \times min(x, y)$
- 这个等于所有点到重心的距离和的两倍
- 证明：
- 以重心为根建树，假设一条边$a-b$, b是a的父亲，则这条边的贡献就是$2 \times a$的子树大小
- $=$所有点深度和$*2$ $=$所有点到重心的距离和$*2$
- 加入一个点，维护所有点到当前重心的距离和

problem D

- 每次加入一个叶子x后，重心最多会向x所在的子树移动一步，根据size判断一下即可

problem D

- 每次加入一个叶子x后，重心最多会向x所在的子树移动一步，根据$size$判断一下即可
- 重心到所有点的距离和可以用树上差分，每加入一个点，就把它到根路径上每个点的权值都$+1$，距离和即为重心到根路径上的权值和

problem D

- 每次加入一个叶子x后，重心最多会向x所在的子树移动一步，根据$size$判断一下即可
- 重心到所有点的距离和可以用树上差分，每加入一个点，就把它到根路径上每个点的权值都$+1$，距离和即为重心到根路径上的权值和
- 时间复杂度$O(n \log n)$

problem E

- 给一棵带边权的树

problem E

- 给一棵带边权的树
- 问有多少个大小恰好为m的点集S满足存在一个点x，$\forall y \in S, dis(x,y) \le k$
  $dis(x,y)$为x,y在树上的距离

problem E

- 给一棵带边权的树
- 问有多少个大小恰好为m的点集S满足
  存在一个点x，$\forall y \in S, dis(x,y) \le k$
  $dis(x,y)$为x,y在树上的距离
- $1 \le n \le 10^5, 1 \le k \le 10^9$，答案对$10^9+7$取模

problem E

- 对于一个合法的点集S，可选的中心点x一定是树上的一个连通块

problem E

- 对于一个合法的点集S，可选的中心点x一定是树上的一个连通块
- 树上的连通块满足点数=边数+1

problem E

- 对于一个合法的点集S，可选的中心点x一定是树上的一个连通块
- 树上的连通块满足点数=边数+1
- 考虑容斥

problem E

- 对于一个合法的点集S，可选的中心点x一定是树上的一个连通块
- 树上的连通块满足点数=边数+1
- 考虑容斥
- 对每个点x，计算一下有多少个大小为m的点集S满足，$\forall y \in S, dis(x,y) \leq k$，加入总答案中

problem E

- 对于一个合法的点集S，可选的中心点x一定是树上的一个连通块
- 树上的连通块满足点数=边数+1
- 考虑容斥
- 对每个点x，计算一下有多少个大小为m的点集S满足，$\forall y \in S, dis(x, y) \leq k$，加入总答案中
- 对每条边(u, v)，计算一下有多少个大小为m的点集S满足，$\forall y \in S, dis(u, y) \leq k, dis(v, y) \leq k$，在总答案中减去

problem E

- 每个点的答案可以通过点分治来求

problem E

- 每个点的答案可以通过点分治来求
- 对一条边(x, y)，假设y是x的父亲
  到x, y距离都$\leq k$的点$=$到x距离$\leq k$的点$-$到x的距离$\leq k$且到y的距离$> k$的点

problem E

- 每个点的答案可以通过点分治来求
- 对一条边(x, y)，假设y是x的父亲
  到x, y距离都$\leq k$的点$=$到x距离$\leq k$的点$-$到x的距离$\leq k$且到y的距离$> k$的点
- 前一部分已经通过点分治求得

problem E

- 每个点的答案可以通过点分治来求
- 对一条边(x, y)，假设y是x的父亲
  到x, y距离都$\leq k$的点$=$到x距离$\leq k$的点$-$到x的距离$\leq k$且到y的距离$> k$的点
- 前一部分已经通过点分治求得
- 后一部分的点只可能在x的子树内，即要求x的子树内，到根距离$> y$到根距离$+k$，且$\leq x$到根距离$+k$的点数
  这个可以通过dfs求得

problem E

- 每个点的答案可以通过点分治来求
- 对一条边(x, y)，假设y是x的父亲
  到x, y距离都$\leq k$的点$=$到x距离$\leq k$的点$-$到x的距离$\leq k$且到y的距离$> k$的点
- 前一部分已经通过点分治求得
- 后一部分的点只可能在x的子树内，即要求x的子树内，到根距离$> y$到根距离$+k$，且$\leq x$到根距离$+k$的点数
  这个可以通过dfs求得
- 时间复杂度$O(n \log n)$

problem F

- 给n个点的树，其中有m个点是关键点

problem F

- 给n个点的树，其中有m个点是关键点
- 问有多少个关键点构成的点集，满足任意两个点在树上的距离都$\leq k$

problem F

- 对于一个点集S，如果任意两个点的树上距离$\leq k$，当且仅当存在一个点x，满足x到点集中的任意一个点的树上距离$\leq \frac{k}{2}$

problem F

- 对于一个点集S，如果任意两个点的树上距离$\leq k$，当且仅当存在一个点x，满足x到点集中的任意一个点的树上距离$\leq \frac{k}{2}$
- 由于k可能是奇数，所以对原树，把每条边也用一个点来表示

problem F

- 对于一个点集S，如果任意两个点的树上距离$\leq k$，当且仅当存在一个点x，满足x到点集中的任意一个点的树上距离$\leq \frac{k}{2}$
- 由于k可能是奇数，所以对原树，把每条边也用一个点来表示
- 原题等价于有多少个关键点构成的点集S，满足在新树中，至少存在一个点，到S中的所有点的距离都$\leq k$

problem F

- 对于一个点集S，如果任意两个点的树上距离$\leq k$，当且仅当存在一个点x，满足x到点集中的任意一个点的树上距离$\leq \frac{k}{2}$
- 由于k可能是奇数，所以对原树，把每条边也用一个点来表示
- 原题等价于有多少个关键点构成的点集S，满足在新树中，至少存在一个点，到S中的所有点的距离都$\leq k$
- 套用上一题做法即可