A

重心唯一等价于以某个点为根，不存在 $size=\dfrac{n}{2}$ 的节点。

不妨先找到原树的重心并以其为根。枚举断哪条边（设断 $x\to fa_x$ 的）（假如这条边两边 size 就是 $\dfrac{n}{2}$ 显然不合法）。那么新连到的地方需要满足祖先 p 的 $size$，没有任何一个满足 $size_p+size_x=\dfrac{n}{2}$ 的。

注意到 $size_p=t$（t 为定值）的 p 互不为祖先关系，那么其实应该加的贡献就是 $size_x\times(n-\sum_{size_y=\frac{n}{2}-size_x}size_y)$。就是 $size_x(n-(\dfrac{n}{2}-size_x)T)$，$T$ 是不在 x 子树内的 $size=\dfrac{n}{2}-size_x$ 的数的个数。dsu 即可。树上差分可以线性。

如果最开始时有两个重心，那连到的点必须在另一边。

总时间复杂度 $O(n)$。

由于 2×10^6 卡不了小常数 $\log$，dsu 等算法可能还是可以过。

由于放在第一题，部分分给得特别良心。

B

$l_i=r_i$ 是经典问题。结论是，定义比较函数 $cmp(a,b)$ 等于 $[a+b>b+a]$（$+$ 是字符串拼接，$>$ 是字典序），那把所有数按照 cmp 排序就是答案。可以证明这样结果最大，且 cmp 满足 `sort` 需要的所有性质。

枚举每个数，算他对答案的贡献。

枚举 $a_i=j$，j 对答案有 $j\times10^k$ 的贡献，当且仅当 j 排完序之后后面有 k 个数位。

设 $cnt(x,y)$ 表示 B_x 中，$cmp(o,j)=0$，且 $len(o)=y$ 的数的个数。（$y=0$ 表示 B_x 中，$cmp(o,j)=1$ 的数的个数）

则转移可以简化为 $f(t-1,k-w)cnt(t,w)\to f(t,k)$。这可以看成多项式相乘：设 $f_t(x)=\sum cnt(t,i)x^i$，则我们要求 $j\sum_u 10^u([x^u]\prod_{t\neq i}f_t(x))$。

分治乘法过于傻了。注意到我们就是求 $jg(10)$（$g(x)=\prod_{t\neq i}f_t(x)$）也就是 $j\prod_{t\neq i}f_t(10)$，直接 $O(n)$ 计算即可。

如果按照 cmp 顺序从小到大计算 f，可以发现问题变为：单点修改，求全局乘积，由于不一定不会乘 0，需要用线段树维护。

注意这个问题用 STL sort 复杂度是错的，需要先做一些处理（比如随机打乱输入，或者预处理二分哈希快速比较）！

C

把每个询问中 $(x,y),(k+1,z)$ 称为关键点。用 LCT 维护关键点的虚树即可。

具体需要写三坨东西：

- 换根的求 k 级祖先、求链上 $size$ 和、所有点 $size,size^2$ 和。（码量约 2.5k）
- 一个 LCT，支持链加链求和，Link，Cut。（码量约 2k）

- 对虚树结构的维护。注意到，新加一个点，只会在虚树上多至多两个点：这个点，以及这个点的最近祖先，使该祖先子树内有除了这个点之外其他已经加入虚树中的点。为了找到这个"最近祖先"，需要对每棵换根后的树维护 dfs 序上面的动态开点线段树，支持单点修改区间求和，并找到区间内最浅的关键点（可以画图理解），每次二分。（码量约 1.5k）每次加入新点时分类讨论（超容易写错，要特别注意下传放在 LCT 上的标记）。（码量约 2k）

总码量约 7k~8k。

部分分就是正解的 part。假如写错了换根或者虚树结构维护得有问题，可以过 $y=z=1$（既不需要换根，也不需要维护虚树结构），或者 $z=1$（不需要换根）。假如直接目标就是这两个部分分，可以省 2~3k 码量。假如不会在线修改虚树结构，可以过 $type=0$ 的点。