7047--【11.23 测试】数数

由调整法，k为偶数时，我们选择的肯定是将数组排序后的前$\frac{k}{2}$与后$\frac{k}{2}$个。k为奇数时，选择的中位数显然不影响答案。直接计算即可。

7048--【11.23 测试】数树

由于不满足这个约束的条件很严格，我们考虑容斥。选择树上一组大小为k的边集并强制这些边不满足条件。可以发现，选择的边中不能有两个边的起点相同或终点相同，既选择的边构成很多条不相交的链。由于确定了链首的权值后整个链的权值就都确定了，所以对这些链与点分配权值使选择的边都不合法（其他边随意）的方案数为$(n-k)!$。因此，我们在树上做一个简单的插头dp，每个点的可能状态为没有选连接的边，选了一条向下/向上指的边，或者选了两条连起来的边。由于阶乘的贡献无法通过乘法合并，所以需要再记录一维背包大小。最后根据状态定义做一个树上背包即可。复杂度$O(n^2)$

7049--【11.23 测试】鼠树

我们考虑对每个黑点打标记，记录加上的权值和w_i。在询问子树和时，如果这个黑点在子树内，那么它管辖的所有点也在子树内，所以我们可以直接对权值乘点数求和。对于剩下的白点，他们显然都受同一个黑点管辖，所以直接求那个黑点的权值即可。对于这部分的点数，我们维护每个黑点管辖的点数就可以求出。

所以，我们只需要维护每个黑点管辖的点数c_i，每个黑点的权值w_i以及两者的乘积。在黑点集合不变的时候，这些都很容易维护。现在考虑加入一个黑点。他管辖的点数可以直接由子树中c_i的和求出，而他的权值可以直接设为这个点之前的归属点的权值。当然，我们还要更新它之前归属点的c_i。

这时我们发现需要动态地求每个点的归属点。一种无脑的方法是用树状数组维护每个点到根节点的黑点数然后直接二分。由于树状数组的常数很小，这个方法可以通过。事实上有一个单log的算法：对树做重链剖分，维护每条链上的黑点按深度排序的集合以及最大最小值。这样我们只需要在一条重链上二分找答案。

最后考虑删除一个点。最麻烦的部分是把这个点的权值保留下来。我们考虑这个点的管辖点的权值与该点权值的差，单独开一个树状数组对这个差做子树加，再用题目中的操作`4`把子树内不影响的点减回去即可。

以上操作都可以用树状数组或线段树完成，复杂度$O(n\log n)$。

7050--【11.23 测试】ckw 的树

算法1

设$E(x)$为从x开始的期望游走时间。
暴力列方程高斯消元$O(n^3)$，可以过subtask1。
如果您有非常强的优化技巧有概率过subtask4。

算法2

对于subtask3，也就是树的形态是一朵菊花。
可以发现只有三种变量：根，被标记叶子，未标记叶子。
列个方程直接解一下就好了。
时间复杂度$O(n)$
可以过subtask3。

算法3

对于subtask2，可以得到一系列连续的五个元素的方程。
可以从后往前用$E(n)$和$E(n-1)$表示出$E(1), E(2), .., E(n-2)$。
然后最后还剩两个方程可以直接解出$E(n)$和$E(n-1)$，然后算出所有的期望。
时间复杂度$O(n)$
或许有什么其它做法也可以过。
可以过subtask2。

算法4

稀疏矩阵消元，可以过subtask1和subtask4。

算法5

设$fa(x)$为x的父亲,$son(x)$为x的儿子集合。
$$sum(x) = \sum_{y \in son(x)} E(y)$$
我们可以用一次dfs，对于每一个结点x，求出四个值a_x, b_x, c_x, d_x
表示$E(x) = a_x E(fa(x)) + b_x E(fa(fa(x))) + c_x sum(fa(x)) + d_x$
然后由于根没有父亲，所以可以再dfs一遍，求出每个点的$E(x)$
我们发现这个sum有点麻烦，对于每一个节点的所有儿子我们可以暴力高斯消元消掉sum，所以时间复杂度$O(nD^3)$(D为最大度数)

可以过subtask1,subtask2,subtask4。

算法6

对于算法5的高斯消元，我们发现这个方程比较特别，直接将所有方程加起来可以解出$sum(fa(x))$，所以就直接解出$sum(fa(x))$然后用代入法解出所有的$E(x)$