

SOI 2019

第一试 solution

A : 小 A 的树

其实是真模板题。

考虑无6操作，简单的树链剖分即可，时间复杂度 $O(q \times \log n)$ 。

考虑换根lca，如何去在log级复杂度内计算出答案。

若求两点u,v的lca，此时根为rt。

设Lca(u,v)为在以重心为根时u,v两点的lca，deep(i)表示i在以重心为根的树的深度。

1) 若u,v两点都在以rt为根的子树内，答案即为Lca(u,v)。

2) 若u,v两点一点在子树内，另一点在子树外，答案即为rt。

3) 若u,v两点都在子树外，答案为Lca(u,rt)与Lca(v,rt)中deep最大的那个。

所以综合起来，只要在Lca(u,v), Lca(u,rt), Lca(v,rt)中取deep最大的那个。

倍增lca与树链剖分保证总时间复杂度 $O(q \times \log n)$ 。

说真的码量其实不大，标程才10k，在思维能力小于NOIP T1难度，并且良心的出题人还发下去可以去对拍的checker.exe。

B : 组合数问题

Question 1 求 $C_{a_1+b_1}^{a_1}$ 。只要会求逆元即可。

Question 2 求 $\sum_{i=1}^n \sum_{j=i+1}^n C_{a_i+a_j+b_i+b_j}^{a_i+b_i}$ 。对于 $C_{a_i+a_j+b_i+b_j}^{a_i+b_i}$ ，与式子 $C_{a_1+b_1}^{a_1}$ 相似。因为 $a_i, b_i \leq$

1000。所以考虑 $C_{a_1+b_1}^{a_1}$ 的其他意思。 $C_{a_1+b_1}^{a_1}$ 在代数意义上代表从(0,0)到 (a_1, b_1) ，只能往上或右走的方案数。所以设 $f(i, j)$ 为走到 (i, j) 的方案数，则 $f(i, j) = f(i, j-1) + f(i-1, j)$ 。所以每个为从(0,0)到 (a_i+b_i, a_j+b_j) 。发现 $a_i+b_i \leq 2000$ ，所以开一个桶记录 a_i+b_i 的次数。然后就暴力一下即可，时间复杂度 $O(4 \times 10^6)$ 。

Question 3 $\sum_{i=1}^n \sum_{j=i+1}^n C_{a_i+a_j+b_i+b_j}^{a_i+a_j}$ 。与Question 2类似，考虑其代数意义是(0,0)到 (a_i+a_j, b_i+b_j) ，因为i与j不在一边所以不好去维护。通过平移可得从 $(-a_i, -b_i)$ 到 (a_j, b_j) 的方案数。在dp上式子为 $f(i, j) = f(i, j-1) + f(i-1, j)$ ，因为是加法所以可以全部得到。就可以容易想到将每个 $f(-a_i, -b_i)++$ ，然后再简单做一下容斥即可，时间复杂度 $O(4 \times 10^6)$ 。