T1

显然题目和权值大小没有关系，考虑先离散化

对于20%的数据，找出B序列中和A序列中值不同的不同数的数量即可

考虑将所有A_i和B_i连一条无向边，现在得到一张m个点的无向图(m为两个序列中总的不同数的个数)

对于一个连通块S，假设有x个点和y条边，那么只需要选出$x-1$条边来构成一棵树的结构（除了根节点之外其他所有值指向其父节点），必然能够使得所有值相同

所以$ans=\sum(\mid S\mid -1)$

答案也等于总点数减去连通块数量，用并查集，dfs，bfs都可以做

时间复杂度：$O(n\log n)$

空间复杂度：$O(n)$

T2

问题就是判断给定图是否能够黑白染色

对于30%的数据，对于每次操作1，暴力给图染色一遍，判断是否可染即可，复杂度$O(q\cdot n)$

对于100%的数据，用可撤销扩展域并查集来做，如果加了一条边，显然这条边连的两个点x,y的颜色不能相同，也即在扩展域并查集中x和$y+n$，$x+n$和y需要放在同一个集合中，如果发现加边之前x和y或者$x+n$和$y+n$已经在同一个集合中的话，那么显然出现了冲突，也即在当前加入的这条边没有被删除之前，之后再怎么加边都是无法黑白染色了，所以问题转化为维护一个可撤销的扩展域并查集，同时记录加入了一条使图不能染色的边之后总共有多少条无用边，复杂度$O(n+q\log n)$

当然也能用LCT来代替并查集，维护加边删边和判断两个点是否在同一棵树中即可

T3

最小的数据，连接起来串最长只有3948，连接之后$O(3948^2)$暴力匹配即可

20%的数据连接起来最长只有485572，用哈希或者KMP优化一下匹配，$O(485572)$

对于40%的数据，假设F_i和t串的匹配数为cnt_i，可以发现每次连接两个串F_i, F_{i-1}之后，$cnt_{i+1} = cnt_i + cnt_{i-1} + x$，其中$x$为$F_i$的长度小于$|t|$的后缀和$F_{i-1}$的长度小于$|t|$的前缀连接起来的串和$t$串匹配位置的数量，所以我们只需要维护$F_i$的长度不超过$|t|-1$的前后缀即可，每次转移只要加上前后缀匹配的数量就好了，复杂度$O(n|t|)$

60%的数据，可以发现当某个F_i的长度大于等于$|t|-1$了之后，之后所有的$F_j, j \geq i$的前缀都是一样的，而所有$F_j, j \geq i$的后缀只会在两种串之间变化，假设两个后缀和前缀连接之后的匹配的数量分别为x和y，那么之后的转移必然是$cnt_i = cnt_{i-1} + cnt_{i-2} + x$或者$cnt_i = cnt_{i-1} + cnt_{i-2} + y$，且两者交替出现，那么得到$x$和$y$之后就可以直接转移了，由于字符串长度增长速度很快，长度很快就能超过$|t|-1$，定一个常数C，先按之前的方法转移，转移到C次之后就可以直接$O(1)$交替转移了，复杂度$O(C|t|+n)$

根据交替转移$\begin{cases} cnt_i = cnt_{i-1} + cnt_{i-2} + x \\ cnt_i = cnt_{i-1} + cnt_{i-2} + y \end{cases}$

得到$cnt_i = 2 \times cnt_{i-2} + cnt_{i-3} + x + y$

那么可以用矩阵快速幂优化，复杂度为$O(C|t| + 4^3 \times \log n)$