NOIP 模拟赛 Day2

A

Keyword:二分，数据结构

出这题的初衷是为了出一道跟上年的 day1 t2 同类型的题目

有一个显然正确的做法就是每次取最大的 c[i]个，当我们取前 c[i]大的时候，我们我们发现会改变相对顺序，而发生相对顺序改变的位置都是第 c[i]大的权值的部分，如果在这一部分，我们取的是这部分的最后几个，那么相对顺序是不会发生改变的，维护这个就好了，比如原本是[4,4,3,3,3,2]，我们现在要取 4 个，按照原来的做法会得到[3,3,2,2,3,2]，如果用后面的做法会得到[3,3,3,2,2,2]，用数据结构维护。

其实可以用二分，二分出答案之后判断可不可行就只要考虑一个这样的贪心：我们先贪心的取最开始前 c[i]大的（不管后来是不是前 c[i]大），如果不够，那么会往后面取，详情见 std

B

Keyword:贪心

考虑 X=0 的时候怎么做，显然是先强制所有都选择 y[i]，然后按照 z[i]-y[i]从大到小排序，然后选择前 Z 大的更改为 z[i]。

那么当 X>0 的时候，我们先强制所有都选择 x[i]，然后将 z[i]-y[i]从大到小排序，枚举一个 z[i]-y[i]的分界值，选择 z[i]都在分界值以前，选择 y[i]的都在分界值以

后，那么可以发现，由于前面已经强制都选择 x[i]，那么现在选择 z[i]的就是分界值之前的三元组里 z[i]-x[i]最大的 Z 个，分界值后的同理。

然后维护这个东西可以直接上数据结构(~~如果常数写得比快排还要优那么可能是可以过的~~）

当然可以不用数据结构（毕竟是~~假~~noip 模拟），首先由于数值的范围都是[0,500000]，那么我们可以桶排，然后，我们在枚举分界点的时候相当于每次加入一个 z[i]-x[i]以及删去一个 y[i]-x[i]，注意到不断增加而不删除就会导致第 Z 大的值是不递减的，不断删除同理，所以可以先排一次序，然后用两个指针扫一下就好了。

时间复杂度 O(n)

~~（真 noip：标程没有数据结构：)~~

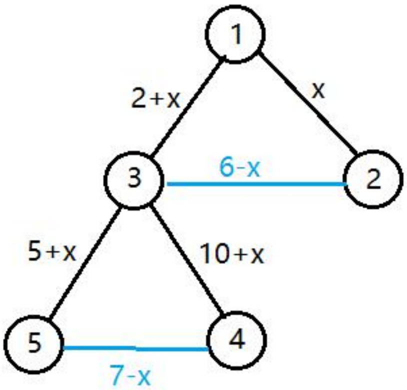
C

Keyword:最小生成树

把图分成正图(k+x)和负图(k-x)，且称正树为正图中的最小生成树，负树为负图中的最小生成树

有个很明显的结论是，无论 x 的值是多少，最后的最小生成树的边一定是正图中的最小生成树上的边或负图的最小生成树上的边(因为正图和负图都是联通图)

当 x 趋近于无穷小的时候，正树就是图的最小生成树，当 x 逐渐增大，负树中的边会逐渐取代正树中的边（但不一定按照从小到大顺序取代），如下图：



当 x>-1.5 时 7-x 的边取代 5+x，当 x>2 的时候 6-x 取代 2+x

但是这样并不影响我们从小到大加入负树中的边，因为上面的情况只会出现在两边产生的环没有边相交的情况，而如果有相交的部分，那么 k 比较小的会先做出决策。

然后就得到如下算法：

先分别做最小生成树算法得到正树和负树，以正树作为初始版本的最小生成树，然后将负树上的边从小到大加进生成树中，对于一条负边<x,y,k>，找出 x 到 y 之间 k 值最大的正边，然后可以算出当前负边取代这条正边的 x 的下界，接着修改树的形态，最后将所有算出的 x 的下界排序，就可以得到每个阶段的最小生成树的权值了。

树边删除增加可以用 LCT 做。

时间复杂度 O(n log n)

由于考虑到是 noip 模拟，所以只要得到结论打了暴力也给了 90 分。

（PS：这题的最后的数据其实造得很水，图是完全随机的，所以树是期望高度为 n 的，可以直接暴力求 lca，暴力翻转边，然后复杂度也是 n log n 的

~~真模拟 NOIP 数据水~~