cf521D shop

题目大意

给出k个能力值$a_1 \dots a_k$，给出n项操作，每项操作形如将某项能力值修改为一个数，或加一个数，或乘一个数。

从中选出至多m项操作，使得依次执行这些操作后，最大化所有能力值之积。输出一个方案。

数据范围

$1 \le k \le 10^5, 0 \le m \le n \le 10^5$.
$1 \le t_j \le 3, 1 \le i_j \le k, 1 \le b_j \le 10^6$.

解题过程

先考虑只有乘的情况，此时只要将所有要乘的数从大到小排序，取前面m个即可。

考虑加。对于一项能力值，先加后乘不会使答案变劣，先加大数字后加小数字也不会使答案变劣。不妨设有t项操作是对a_1进行加操作，第i项操作加的数字为$b_i, b_i > b_{i+1}(1 \le i < n)$。则第$i$项操作等效于乘$\frac{a_1+\sum_{j=1}^{i} b_i}{a_1+\sum_{j=1}^{i-1} b_i}$。接下来按照只有乘的情况处理即可。

考虑赋值。对于一项能力值，必然存在一种最优方案只进行一次赋值，且赋值应当最先进行。首先对于同一项能力值的赋值只应该保留最高的一个，然后可以发现这次赋值其实等效于一次加操作（赋值为b等价于加$b-a$(a表示初始值)，接下来按照只有乘和加的情况处理即可。

按照上面所说，只要先把所有操作转化为乘，按照乘数大小排序，取有效的前m项，再按照操作类型排序，依次输出即可。

还有一个问题，排序时需要涉及分数比较，分子分母都可能高达10^{11}，用乘法比较时中间值会高达10^{22}，超出c++中long long的表示范围。对此，可以发现分子和分母相差最多一项，于是可以将所有的分子减去其分母，这样相当于所有数减去1，排序结果不变，但分子不会超过10^6，用乘法比较时中间值不会超过10^{18}，不会超出long long的表示范围。
总时间复杂度$O(nlogn)$.

agc036_f Square Constraints

题目大意

给出一个整数N。对$(0, 1 ... 2N-1)$进行重新排列得到$(P_0, P_1 \dots P_{2N-1})$。求满足下列条件的排列数量：

$$N^2 \le i^2 + P_i^2 \le (2N)^2, \forall 0 \le i \le 2N-1$$

答案可能很大，输出对M取模的结果。

数据范围

$1 \le N \le 250$
$2 \le M \le 10^9$

解题过程

容易发现每个P_i的取值范围是一个区间$[l_i, r_i]$,且l_i, r_i均单调不上升。当$i >= N$时，$l_i = 0$。

于是可以看成求满足2*N个限制的排列的数量。如果限制只有上界没有下界，那么只要将所有数按上界排序，然后从小到大依次考虑每个上界即可。设排序后第i个上界（从1开始标号）为U_i，则其贡献为$U_i + 1 - (i - 1)$。现在加入下界，考虑容斥。

设对所有i有$P_i \le r_i$,并枚举k个i满足$P_i < l_i$的方案数为F_k。由容斥原理可得答案为$F_0 - F_1 + F_2 \ldots$

如何求F_k？

将对于$i = 0...N-1$，以l_i为第一关键字，r_i为第二关键字；对于$i = N...2N-1$，以r_i为第一关键字，l_i为第二关键字，按关键字从小到大排序得到序列$q[1...2N]$。

k确定时，设$f[i][j]$表示考虑序列q前i项，所有项满足$P_i < r_i$，且强制了j项满足$P_i < l_i$的对答案的贡献（这里的贡献指考虑所有数之后按限制条件排序后相乘时的贡献）。

转移时，设前$i-1$项中比N小的有t项。考虑第i项，若$q[i] >= n$，则

$$f[i][j] = f[i-1][j] * (r_{q[i]} + 1 - (i - 1 - (t - j)))$$

否则

$$f[i][j] = f[i-1][j] * (r_{q[i]} + 1 - (2N - 1 - q[i] + (k - j))) + f[i-1][j-1] \\ * (l_{q[i]} + 1 - (i - 1 - (t - (j - 1))))$$

总时间复杂度$O(n^3)$.

agc028_c Min Cost Cycle

题目大意

给出一个N个点的有向图，边有边权。每个点有两个权值，第i号点的两个权值分别是A_i, B_i。

对于所有点对(x, y)，存在从x连向y的边，它的权值是$min(A_x, B_y)$。

求一个环，经过图中的每个点恰好一次，且经过的边的权值和最小。输出这个最小的权值和。

数据范围

$2 \le N \le 10^5$
$1 \le A_i, B_i \le 10^9$

解题过程

先把所有A_i和B_i放在一起排序。显然答案的一个下界是排序后的前N项之和。

考虑什么时候可以取到这个下界。

引理：如果存在一个i，A_i和B_i在前N项中全部出现，则存在一个环的答案为前N项之和。

证明：根据在前N项中的出现情况，我们把点分为4类：
1.A和B都出现；
2.只有A出现；
3.只有B出现；

4.A和B都不出现。
把所有2类点连接成一条链，设为$T_{2,head}->...->T_{2,tail}$.
把所有3类点连接成一条链，设为$T_{3,head}->...->T_{3,tail}$.
1类点和4类点的数量必定相同，且都存在。设各有m个，并记为$T_{1,1...m},T_{4,1...m}$.
则这样的环满足条件：

$$T_{2,head}->...->T_{2,tail}->T_{4,m}->T_{3,head}->...->T_{3,tail}->T_{1,1}->T_{4,1}-\\>T_{1,2}->T_{4,2}->...->T_{1,m}->T_{2,head}$$

.
证毕。

如果每个i在前N项中出现恰好一次，再分情况讨论：
1.如果前N项的类型（A或B）全都相同，则对于任意一个满足条件（经过所有点恰好一次）的环，必定取到这个下界。实际上是所有点都是2类点（或3类点），直接连接成一个环。
2.如果前N项的类型不全相同，必定无法取到这个下界。因为2类点和3类点直接相连时，它们之间的边必然无法满足条件。

此时，把前$N-1$项与第$N+1$项的和是答案的又一个下界。能取到这个下界当且仅当前$N-1$项与第$N+1$项满足引理的条件，或类型全都相同。

如果仍然不能取到，则备选答案变为前$N-1$项与第$N+2$项的和，或前$N-2$项与第N项与第$N+1$项的和。注意到第N项与第$N+1$项对应的i必然相同，于是这两个备选答案必然都满足引理条件，于是可以取到。取其中较小值即可。

时间复杂度$O(nlogn)$.