Solution

dy0607

January 11, 2018

1 Drink

Source: Codeforces Round #223 Div1 E

1.1 25%

对于一个给定的区间$[l,r]$，怎样操作才是最优的呢？

显然我们不会对一个数操作两次．设倒数第i次操作中选中的数为A_i，那么最终得到的水量为V为：

$$V = \sum_{i=1}^{r-l+1} \frac{A_i}{2^i}$$

因此我们一定会把水量较多的留到最后操作，那么我们只需要区间内排个序统计就可以算出答案，暴力是$O(n^3 \log n)$的．

1.2 40%

注意到题目输出实数，对于上式中较大的i，$\frac{A_i}{2^i}$可以忽略不计．

这样我们只需要找到一个区间内最大的大约$T = 50$个数就可以了，在固定左端点的情况下向右移动右端点，并维护最大的T个数即可．$O(n^2T)$

1.3 100%

考虑分开计算每个数的贡献，对于第i个数和一个区间$[l,r]$，这个数的贡献只与这个区间内比它大的数的个数t有关．与上面类似的，当$t > T$时，贡献可以忽略不计．

于是我们在i的左边和右边分别找T个离i尽量近且比w_i大的数，假设左边找到的数的位置从右往左是$l_1, l_2, ..., l_T$，右边从左往右是$r_1, r_2, ..., r_T$，方便起见$l_0 = r_0 = i$. 则i的贡献sum_i可以估算为：

$$\begin{aligned} sum_i &= w_i \times \sum_{u=1}^{T}\sum_{v=1}^{T}(l_{u-1} - l_u) \times (r_v - r_{v-1}) \times \frac{1}{2^{u+v-1}} \\ &= 2w_i(\sum_{u=1}^{T}\frac{l_{u-1} - l_u}{2^u})(\sum_{v=1}^{T}\frac{r_v - r_{v-1}}{2^v}) \end{aligned}$$

现在唯一的问题是找到最近的T个大于w_i的数，一种高效的方法是按数从小到大的顺序计算贡献，用链表维护比当前数大的数．计算完一个数的贡献后，将它从链表中删去.

复杂度为$O(nT + n \log n)$

2 Tournament

`Source: CS Acadamy - Tournament Cycle`

2.1 k = 3

通过~~猜结论~~观察，如果图中没有大小为3的环，那么图一定为DAG，而DAG数量为$n!$.

Lemma 1 任意给出一个拓扑序，能且仅能构造出一个没有环的竞赛图.

Proof 1 一个DAG图一定存在至少一个入度为1的点，而在竞赛图中，不可能存在两个入度为1的点，因为点两两之间都有连边. 这使得无环竞赛图的拓扑序是唯一确定的. 对于节点序列$V_1, V_2, ..., V_n$，当且仅当对于任意$i < j$，存在边(V_i, V_j)时，竞赛图的拓扑序为$V_1, V_2, ..., V_n$.

2.2 k = n

要解决接下来的任务，需要先猜证明出一个重要的结论. ~~或者直接打表查询OEIS~~.

Lemma 2 竞赛图中，一个$n(n \geq 3)$个点的强连通分量中一定包含有长度为$i(3 \leq i \leq n)$的简单环.

Proof 2 不妨采用归纳证明，在$n = 3$时该结论显然成立，$n = 1$时也可以看作成立.

否则考虑强连通分量中的任意一个点S，如果将这个点删去，可能产生若干个新的强连通分量. 将这些分量视为一些点，则根据之前的定理，它们的拓扑序是确定的，且拓扑序在前的强连通分量内的所有点都向拓扑序在后的所有点有连边. 设按拓扑序排序后的强连通分量为$G_1, G_2, ..., G_m$.

考虑这样构造出一个大小为n的环：$S \to G_1 \to G_2 ... G_m \to S$，如果这个结论对较小的强连通分量成立，那么我们在任意一个分量G_i中，从任意一个点开始一定可以找到一条路径遍历其中的所有点. 又因为S到G_1以及G_m到S都一定至少有一条边（否则整个图不强连通），所以一定可以找到一个这样的构造.

然后考虑大小小于n的环. 如果$m = 1$(即包含一个$n - 1$的强连通分量)，那么可以归纳到$n - 1$的情况. 否则，由于拓扑序靠前强连通分量中的所有点向靠后的所有点都有连边，我们可以在大小为n的环的基础上任意跳过一些点，不难发现可以构造出任意大小的环.

$k=n$时，问题就是有多少n个点的强连通竞赛图，设这个答案为$f(n)$，考虑转化为求有多少n个点的竞赛图不强连通，这只需要枚举拓扑序最靠前的强连通分量的大小：(为了方便设$g(n)=2^{\binom{n}{2}}$．)

$$f(n)=g(n)-\sum_{i=1}^{n-1} g(n-i)\times f(i)\times\binom{n}{i}$$

$$\frac{f(n)}{n!}=\frac{2g(n)}{n!}-\sum_{i=0}^{n-1}\frac{g(n-i)}{(n-i)!}\times\frac{f(i)}{i!}$$

考虑它们的生成函数：

$$F(x)=\sum_{i=0}^{\infty}\frac{f(i)}{i!}x^i, G(x)=\sum_{i=1}^{\infty}\frac{g(i)}{i!}x^i$$

那么

$$F=2G-F\times G+1\Rightarrow F=2-\frac{1}{G+1}$$

使用多项式逆元可以在$O(n\log n)$时间内计算．

2.3 100%

现在问题转化为：求有多少个大小为n的竞赛图，其中至少包含一个大小至少为k的强连通分量．同样采用补集思想，计算有多少只包含大小小于k的强连通分量的竞赛图, 设为$s(n)$，那么：

$$s(n)=\sum_{i=1}^{min(n,k-1)} s(n-i)\times f(i)\times\binom{n}{i}$$

这个式子同样可以多项式求逆做到$O(n\log n)$，于是问题全部解决．

事实上由于上式转化后是一个线性常系数递推式，复杂度还可以做到$O(k\log k\log n)$，出于~~出题人很懒~~降低代码复杂度的考虑没有出出来．

3 Segment

其实这才是第一题$>_>$.

3.1 100%

注意到$n > m$时，不存在合法的操作序列，答案为0. 又因为有$nm \le 10^5$的条件，可以认为$n < 320$.

我们可以将区间$[l, r)$看成一对括号（左括号在l，右括号在r），位置i最终的值就是这个位置左边的左括号数量减去右括号数量.

设$dp[i][l][r]$表示在位置i，左边有l个左括号，r个右括号的方案数，$f[i][l][r]$表示这些方案当前的贡献和. 每次先令$f[i][l][r] \mathrel{+}= dp[i][l][r] \times (l-r)^k$，再进行转移即可. 转移只需要讨论$i+1$上是否放左括号以及右括号.

$O(n^2m)$