1 排序问题

考虑怎么计算一个长为 len 的序列 a 的 Gobo sort 的期望次数。

令 b_i 表示 a 中 i 出现的次数。

那么期望次数 $E = len!\prod_i \frac{1}{b_i!}$ 。

我们令 Δ_i 其中 $l \le i \le r$ 表示 第 i 个数字添加了几个。

现在我们就是希望 $\prod_{i<l,i>r} \frac{1}{b_i!} \prod_{l\le i\le r} \frac{1}{(b_i+\Delta_i)!}$ 最大。

注意到如果 $\Delta_x > 0, \Delta_y \ge 0, b_x + \Delta_x > b_y + \Delta_y$, 那么我们令 $\Delta_x \leftarrow \Delta_x - 1, \Delta_y \leftarrow \Delta_y + 1$, 答案不会变劣。

从这个结论中，我们可以发现，我们增加的一定是出现次数最少的那些数字，并且我们会让被增加的数字的出现次数尽量平均，也就是存在一个 x ，将所有出现次数不超过 x 的变成 x ，然后将一些 x 变成 $x+1$ 。

这个 x 是可以二分的，二分找到 x 计算答案即可。

时间复杂度 $O(m + \sum n\log(n+m))$ ，其中 $O(m)$ 表示预处理阶乘。

2 游戏

我们称一个数字 x 在 $[l,r]$ 中是好的当且仅当 $\forall d|x, d<x$ 满足 $d<l$, 也就是 $[l,r]$ 中没有 x 除了自己以外的约数。

注意到一个编号是好的数字的那些玩家不会被任何其他玩家攻击，因此游戏要结束必须满足所有编号是好的数字的玩家被改造。

反过来，如果所有编号是好的数字的玩家都被改造，那么每个玩家都会被攻击，游戏会结束。

因此游戏结束的轮数就是最后一个编号是好的数字的玩家被改造的轮数。

我们先考虑一个问题: 在 n 个数字的所有排列中，有 m 个关键的数字，计算所有关键的数字最后一个出现的位置的期望。

由于最大值比较难算，把序列对称一下，考虑算最小值，因此期望就是对所有的 i ，计算前 i 个位置没有关键数字的排列个数求和。

这里把式子列出来:

$$
\begin{aligned}
Ans &= \sum_{perm} last\ position\ of\ key\ number && (1)\\
&= \sum_{perm} n+1-first\ position\ of\ key\ number && (2)\\
&= (n+1)! - \sum_{i=0}^{n} \binom{n-i}{m}(n-m)!m! && (3)\\
&= (n+1)! - (n-m)!m! \sum_{i=0}^{n} \binom{n-i}{m} && (4)\\
&= (n+1)! - (n-m)!m! \sum_{i=0}^{n} \binom{i}{m} && (5)\\
&= (n+1)! - (n-m)!m! \binom{n+1}{m+1} && (6)\\
&= (n+1)!(1-\frac{1}{m+1}) && (7)
\end{aligned}
$$

在这个问题中, $n=r-l+1$, m 就是好的数字的个数。我们发现答案只和好的数字有几个有关，和具体是哪几个无关。

考虑怎么计算好的数字个数, 注意到一个数字 x 是好的当且仅当 $l \le x \le r, x/minp(x) < l$ 。

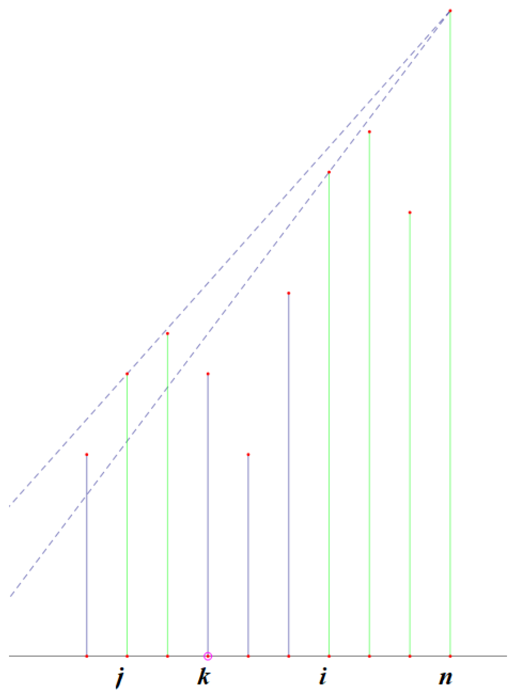
使用欧拉筛法，计算所有数字最小的质因子即可。

时间复杂度 $O(r)$ 。

3 守卫

先考虑计算 $[1,n]$ 这个区间的答案。

n 上一定有保镖。不能被 n 照到的亭子是很多个区间。



先看最右边那个区间，需要在 $[i,n]$ 里面选几个点来（可能是 0 个）照亮它们。可以发现 (i,n) 里面的亭子一定不优，因为这里的亭子能看到的，n 一定也能看到。那么 (i,n) 肯定不选，然后 i 和 $i-1$ 里一定有一个要选。

然后看看 $[k,i]$ 里面的决策会不会影响 $[1,k)$，发现不会，因为 $[1,k)$ 里面的点如果能被 $[k,i]$ 看到，那么肯定会被 n 看到。

所以可以单独把 $[k,i]$ 拿出来决策。

这样可以区间 DP，设 $dp_{1,n}$ 表示 $[1,n]$ 自给自足的答案是多少。那么 n 一定要放，照不到的做成区间 $[l_i,r_i]$ 以后，每个区间的答案是 $\min(dp_{l_i,r_i}, dp_{l_i,r_i+1})$。分别表示上图中选取 $i-1$ 和 i 的情况。

这样复杂度 $O(n^3)$。不难发现如果区间 DP 的时候固定右端点，左端点从右往左扫过去，复杂度可以优化到 $O(n^2)$。

然后发现我们可以顺便计算出所有区间的 dp 值，就做完了。