1 控制人偶

我们可以算出一整个命令串该人偶可以到达的位置，然后重复这个过程直接在坐标上乘就可以了。最后 $T\%|S|$ 的部分再单独处理一下即可。

2 复杂度计算

2.1 60 分做法

如果固定了矩形的左上角（也就是 i 和 j 的枚举），那么剩下的就可以看成是一些共用一个角落的矩形的面积和。

设最大的矩形是 $N \times M$ 的，那么

$$S(N,M)=\sum_{x=1}^{N}\sum_{y=1}^{M}xy=\sum_{x=1}^{N}x\sum_{y=1}^{M}y=\frac{N(N+1)}{2}\cdot\frac{M(M+1)}{2}$$

答案就是 $\sum_{i=1}^{n}\sum_{j=1}^{m}S(n-i+1,m-j+1)$。

逆元可以预处理，时间复杂度 $\mathcal{O}(n^2)$。

2.2 80 分做法

注意到答案也等于 $\sum_{i=1}^{n}\sum_{j=1}^{m}S(i,j)$，展开后可以得到

$$\frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{m}i(i+1)\cdot j(j+1)=\frac{1}{4}\sum_{i=1}^{n}i(i+1)\sum_{j=1}^{m}j(j+1)$$

两部分形式相同，转化为求 $f(n)=\sum_{i=1}^{n}i(i+1)=\sum_{i=1}^{n}i^2+\sum_{i=1}^{n}i$。

答案就是 $\frac{1}{4}f(n)f(m)$，时间复杂度 $\mathcal{O}(n)$。

2.3 100 分做法

利用等式

$$\sum_{i=1}^{n}i^2=\frac{1}{6}n(n+1)(2n+1),\sum_{i=1}^{n}i=\frac{1}{2}n(n+1)$$

可以 $\mathcal{O}(1)$ 处理。

3 复印任务

首先容易知道第 i 个复印机启动的时间是第 $i-1$ 个复印件好的时候。于是乎我们可以使用模拟前 i 个复印件的方式知道每个复印机启动的时间。这个我们用一个堆维护即可 $O(n\log n)$。

然后我们可以发现对于每个询问我们进行二分时间，然后统计这个时间段内能产生多少复印件即可。但是直接二分复杂度并不优。

紧接着我们注意到 $1 \le t_i \le 10^3$。我们可以对同类时间的复印机进行预处理。由于我们知道每个复印件的启动时间，所以我们可以维护三个值：$c[t]$ 代表周期为 t 的复印机的个数，$f[t][p]$ 表示周期为 t 的复印机，且它工作的起始时间对 t 取模以后是 p 的个数（同时我们用 g 表示 f 第二维的后缀和）。同时记录 $s[t]$ 表示所有周期为 t 时间的复印机进入打印状态所需要的 t 的时间周期有多少个。这样我们可以很快统计同类周期的复印机能在一段时间内打印的个数。

假设现在你拥有的时间为 T, 你枚举的复印机的时间周期为 t, 则你能产生的复印件个数为 $c[t] * (T/t) - s[t] - g[t][T\%t+1]$（这个考虑一下细节就知道对不对了）。把所有不同周期的加起来就可以获得答案。

整体复杂度为 $O(m \max t_i \log T)$.

4 A+B Problem

一个非常裸的（卡常数）数据结构题。

显然使用二维树状数组。

这里注意到需要用到二维数组的矩阵加，和矩阵减。

首先我们设原矩阵为 $a_{i,j}$，那我们对其进行二维差分则有 $b_{i,j} = a_{i,j} - a_{i-1,j} - a_{i,j-1} + a_{i,j}$。然后同时也有 $a_{i,j} = \sum_{k=1}^{i} \sum_{l=1}^{j} b_{i,j}$。

然后我们对一个矩阵的修改操作直接可以在 b 数组上面体现，比如对矩阵 $(i,j)-(k,l)$ 进行 $+x$，直接 $b_{i,j} += x, b_{k+1,j} -= x, b_{i,l+1} -= x, b_{k+1,l+1} += x$。

然后同理我们要求的矩阵和可以转化为求前缀和 $S_{n,m}$, 矩阵 $(i,j)-(k,l)$ 进行求和调用 $S_{k,l} - S_{k,j-1} - S_{i-1,l} + S_{i-1,j-1}$ 即可.

$$
\begin{aligned}
S_{n,m} &= \sum_{i=1}^{n} \sum_{j=1}^{m} a_{i,j} \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{i} \sum_{l=1}^{j} b_{k,l} \\
&= \sum_{k=1}^{n} \sum_{l=1}^{m} \sum_{i=k}^{n} \sum_{j=l}^{m} b_{k,l} \\
&= \sum_{k=1}^{n} \sum_{l=1}^{m} b_{k,l} \times (n-k+1)(m-l+1) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} b_{i,j} \times (n-i+1)(m-j+1) \\
&= (\sum_{i=1}^{n} \sum_{j=1}^{m} b_{i,j}) \times (n+1)(m+1) - \sum_{i=1}^{n} \sum_{j=1}^{m} i b_{i,j} \times (m+1) - \sum_{i=1}^{n} \sum_{j=1}^{m} j b_{i,j} \times (n+1) + \sum_{i=1}^{n} \sum_{j=1}^{m} ij b_{i,j}.
\end{aligned}
$$

最终维护四个树状数组的前缀和 $b_{i,j}, ib_{i,j}, jb_{i,j}, ijb_{i,j}$ 即可。

时间 $O(Q \log^2 n)$，注意存放的时候用 char 即可。