联赛题选讲与技巧选讲

ExfJoe

福建省长乐第一中学

September 28, 2016

Part I

联赛题选讲

Outline

NOIP2011 铺地毯

NOIP2011 铺地毯

- 按顺序给出 n 块地毯，每块地毯是个矩形，且按输入顺序覆盖在地面上。给定一个坐标 (x, y)，问在 (x, y) 这个点最上方的地毯是哪一个

NOIP2011 铺地毯

- 按顺序给出 n 块地毯，每块地毯是个矩形，且按输入顺序覆盖在地面上。给定一个坐标 (x, y)，问在 (x, y) 这个点最上方的地毯是哪一个
- $0 \leq n,$ 坐标范围 $\leq 10^5$

NOIP2011 铺地毯

- 按顺序给出 n 块地毯，每块地毯是个矩形，且按输入顺序覆盖在地面上。给定一个坐标 (x, y)，问在 (x, y) 这个点最上方的地毯是哪一个
- $0 \le n$, 坐标范围 $\le 10^5$
- 直接用数组模拟内存开不下

NOIP2011 铺地毯

- 按顺序给出 n 块地毯，每块地毯是个矩形，且按输入顺序覆盖在地面上。给定一个坐标 (x, y)，问在 (x, y) 这个点最上方的地毯是哪一个
- $0 \le n,$ 坐标范围 $\le 10^5$
- 直接用数组模拟内存开不下
- 询问点只有一个且在所有地毯铺完之后

NOIP2011 铺地毯

- 按顺序给出 n 块地毯，每块地毯是个矩形，且按输入顺序覆盖在地面上。给定一个坐标 (x, y)，问在 (x, y) 这个点最上方的地毯是哪一个
- $0 \le n$, 坐标范围 $\le 10^5$
- 直接用数组模拟内存开不下
- 询问点只有一个且在所有地毯铺完之后
- 倒着枚举地毯，答案即为第一个覆盖询问点的地毯

Outline

NOIP2011 选择客栈

NOIP2011 选择客栈

- 一个长度为 n 的区间，每个位置上有颜色 c_i 以及代价 v_i，现在问有多少个区间 $[l, r]$ 满足 $c_l = c_r$ 且 $\min_{l \le i \le r} v_i \le p$

NOIP2011 选择客栈

- 一个长度为 n 的区间，每个位置上有颜色 c_i 以及代价 v_i，现在问有多少个区间 $[l, r]$ 满足 $c_l = c_r$ 且 $\min_{l \le i \le r} v_i \le p$
- $n \le 2 \cdot 10^5$, $1 \le c_i \le 50$, $0 \le p, v_i \le 100$

NOIP2011 选择客栈

- 一个长度为 n 的区间，每个位置上有颜色 c_i 以及代价 v_i，现在问有多少个区间 $[l,r]$ 满足 $c_l = c_r$ 且 $\min\limits_{l \le i \le r} v_i \le p$
- $n \le 2 \cdot 10^5$, $1 \le c_i \le 50$, $0 \le p, v_i \le 100$
- 由于区间端点要求颜色相同，考虑做颜色的前缀和 $sum_{i,j}$ 表示前 i 个位置中颜色 j 的数量

NOIP2011 选择客栈

- 一个长度为 n 的区间，每个位置上有颜色 c_i 以及代价 v_i，现在问有多少个区间 $[l, r]$ 满足 $c_l = c_r$ 且 $\min\limits_{l \le i \le r} v_i \le p$
- $n \le 2 \cdot 10^5$ ，$1 \le c_i \le 50$，$0 \le p, v_i \le 100$
- 由于区间端点要求颜色相同，考虑做颜色的前缀和 $sum_{i,j}$ 表示前 i 个位置中颜色 j 的数量
- 设 $next_i$ 表示最小的满足 $\min\limits_{i \le j \le next_i} v_j \le p$ 的值

NOIP2011 选择客栈

- 一个长度为 n 的区间，每个位置上有颜色 c_i 以及代价 v_i，现在问有多少个区间 $[l, r]$ 满足 $c_l = c_r$ 且 $\min\limits_{l \le i \le r} v_i \le p$
- $n \le 2 \cdot 10^5$, $1 \le c_i \le 50$, $0 \le p, v_i \le 100$
- 由于区间端点要求颜色相同，考虑做颜色的前缀和 $sum_{i,j}$ 表示前 i 个位置中颜色 j 的数量
- 设 $next_i$ 表示最小的满足 $\min\limits_{i \le j \le next_i} v_j \le p$ 的值
- $next_i$ 可以通过倒推得到

NOIP2011 选择客栈

- 一个长度为 n 的区间，每个位置上有颜色 c_i 以及代价 v_i，现在问有多少个区间 $[l,r]$ 满足 $c_l = c_r$ 且 $\min_{l \le i \le r} v_i \le p$
- $n \le 2 \cdot 10^5$, $1 \le c_i \le 50$, $0 \le p, v_i \le 100$
- 由于区间端点要求颜色相同，考虑做颜色的前缀和 $sum_{i,j}$ 表示前 i 个位置中颜色 j 的数量
- 设 $next_i$ 表示最小的满足 $\min_{i \le j \le next_i} v_j \le p$ 的值
- $next_i$ 可以通过倒推得到
- $ans = \sum_{i=1}^{n} sum_{n,c_i} - sum_{next_i - 1, c_i}$

Outline

NOIP2011 计算系数

NOIP2011 计算系数

- 给定一个多项式 $(ax+by)^k$，求多项式展开后 $(x^n)\cdot(y^m)$ 这一项的系数，答案对 10007 取模

NOIP2011 计算系数

- 给定一个多项式 $(ax+by)^k$，求多项式展开后 $(x^n)\cdot(y^m)$ 这一项的系数，答案对 10007 取模
- $0 \le k \le 1000$，$0 \le n, m \le k$且$n+m=k$，$a, b \le 10^6$

NOIP2011 计算系数

- 给定一个多项式 $(ax+by)^k$，求多项式展开后 $(x^n)\cdot(y^m)$ 这一项的系数，答案对 10007 取模
- $0 \le k \le 1000$，$0 \le n, m \le k$且$n+m=k$，$a, b \le 10^6$
- 二项式展开：

$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} \cdot a^i \cdot b^{n-i}$$

NOIP2011 计算系数

- 给定一个多项式 $(ax+by)^k$，求多项式展开后 $(x^n)\cdot(y^m)$ 这一项的系数，答案对 10007 取模
- $0 \le k \le 1000$, $0 \le n, m \le k$且$n + m = k$, $a, b \le 10^6$
- 二项式展开：

$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} \cdot a^i \cdot b^{n-i}$$

- $$ans = \binom{k}{n} \cdot a^n \cdot b^m$$

NOIP2011 计算系数

- 给定一个多项式 $(ax+by)^k$，求多项式展开后 $(x^n)\cdot(y^m)$ 这一项的系数，答案对 10007 取模
- $0 \le k \le 1000$, $0 \le n, m \le k$且$n+m=k$, $a, b \le 10^6$
- 二项式展开：
$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} \cdot a^i \cdot b^{n-i}$$
- $$ans = \binom{k}{n} \cdot a^n \cdot b^m$$
- $\binom{i}{j} = \binom{i-1}{j} + \binom{i-1}{j-1}$

NOIP2011 计算系数

- 给定一个多项式 $(ax+by)^k$，求多项式展开后 $(x^n)\cdot(y^m)$ 这一项的系数，答案对 10007 取模
- $0 \le k \le 1000$, $0 \le n, m \le k$且$n+m=k$, $a, b \le 10^6$
- 二项式展开：
$$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} \cdot a^i \cdot b^{n-i}$$
- $$ans = \binom{k}{n} \cdot a^n \cdot b^m$$
- $\binom{i}{j} = \binom{i-1}{j} + \binom{i-1}{j-1}$
- $O(k^2)$

Outline

NOIP2012 国王游戏

NOIP2012 国王游戏

- 给定某个值 S 与 n 个二元组 (a_i, b_i)，要求重新排列它们，最小化

$$\max_{1\le i\le n}\left\lfloor \frac{S\cdot\prod\limits_{1\le k<i} a_k}{b_i}\right\rfloor$$

NOIP2012 国王游戏

- 给定某个值 S 与 n 个二元组 (a_i, b_i)，要求重新排列它们，最小化

$$\max_{1\le i\le n}\left\lfloor \frac{S\cdot \prod_{1\le k<i} a_k}{b_i} \right\rfloor$$

- $n \le 1000$，$0 < a, b < 10000$

NOIP2012 国王游戏

- 给定某个值 S 与 n 个二元组 (a_i, b_i)，要求重新排列它们，最小化

$$\max_{1\le i\le n}\left\lfloor \frac{S\cdot \prod\limits_{1\le k<i} a_k}{b_i} \right\rfloor$$

- $n \le 1000$, $0 < a, b < 10000$
- 考虑任意调整解的相邻两个位置 $i, i+1$，显然其他元素对答案贡献不变

NOIP2012 国王游戏

- 给定某个值 S 与 n 个二元组 (a_i, b_i)，要求重新排列它们，最小化

$$\max_{1\le i\le n}\left\lfloor \frac{S\cdot \prod\limits_{1\le k<i} a_k}{b_i} \right\rfloor$$

- $n \le 1000$，$0 < a, b < 10000$
- 考虑任意调整解的相邻两个位置 $i, i+1$，显然其他元素对答案贡献不变
- $$\left\lfloor S * \prod_{1\le k<i} a_k * \max\left(\frac{a_i}{b_{i+1}}, \frac{1}{b_i}\right)\right\rfloor$$

NOIP2012 国王游戏

- 给定某个值 S 与 n 个二元组 (a_i, b_i)，要求重新排列它们，最小化

$$\max_{1\le i\le n}\left\lfloor \frac{S\cdot \prod\limits_{1\le k<i} a_k}{b_i} \right\rfloor$$

- $n \le 1000$，$0 < a, b < 10000$
- 考虑任意调整解的相邻两个位置 $i, i+1$，显然其他元素对答案贡献不变
- $$\left\lfloor S * \prod_{1\le k<i} a_k * \max\left(\frac{a_i}{b_{i+1}}, \frac{1}{b_i}\right) \right\rfloor$$
- $$\left\lfloor S * \prod_{1\le k<i} a_k * \max\left(\frac{a_{i+1}}{b_i}, \frac{1}{b_{i+1}}\right) \right\rfloor$$

NOIP2012 国王游戏

NOIP2012 国王游戏

- 两个式子只有 max 部分不同，观察这部分式子

NOIP2012 国王游戏

- 两个式子只有 max 部分不同，观察这部分式子
- 只看第一项，$\frac{a_i}{b_{i+1}}$ 与 $\frac{a_{i+1}}{b_i}$，则解变优应有 $a_{i+1} \cdot b_{i+1} < a_i \cdot b_i$

NOIP2012 国王游戏

- 两个式子只有 max 部分不同，观察这部分式子
- 只看第一项，$\frac{a_i}{b_{i+1}}$ 与 $\frac{a_{i+1}}{b_i}$，则解变优应有 $a_{i+1} \cdot b_{i+1} < a_i \cdot b_i$
- 若不是第一项最大，那么我们发现：

NOIP2012 国王游戏

- 两个式子只有 max 部分不同，观察这部分式子
- 只看第一项，$\frac{a_i}{b_{i+1}}$ 与 $\frac{a_{i+1}}{b_i}$，则解变优应有 $a_{i+1} \cdot b_{i+1} < a_i \cdot b_i$
- 若不是第一项最大，那么我们发现：
- $$\frac{1}{b_i} \le \frac{a_{i+1}}{b_i} \le \max\left(\frac{a_{i+1}}{b_i}, \frac{1}{b_{i+1}}\right)$$

NOIP2012 国王游戏

- 两个式子只有 max 部分不同，观察这部分式子
- 只看第一项，$\frac{a_i}{b_{i+1}}$ 与 $\frac{a_{i+1}}{b_i}$，则解变优应有 $a_{i+1} \cdot b_{i+1} < a_i \cdot b_i$
- 若不是第一项最大，那么我们发现：
- $$\frac{1}{b_i} \leq \frac{a_{i+1}}{b_i} \leq \max\left(\frac{a_{i+1}}{b_i}, \frac{1}{b_{i+1}}\right)$$
- $$\frac{1}{b_{i+1}} \leq \frac{a_i}{b_{i+1}} \leq \max\left(\frac{a_i}{b_{i+1}}, \frac{1}{b_i}\right)$$

NOIP2012 国王游戏

- 两个式子只有 max 部分不同，观察这部分式子
- 只看第一项，$\frac{a_i}{b_{i+1}}$ 与 $\frac{a_{i+1}}{b_i}$，则解变优应有 $a_{i+1} \cdot b_{i+1} < a_i \cdot b_i$
- 若不是第一项最大，那么我们发现：
- $$\frac{1}{b_i} \le \frac{a_{i+1}}{b_i} \le \max\left(\frac{a_{i+1}}{b_i}, \frac{1}{b_{i+1}}\right)$$
- $$\frac{1}{b_{i+1}} \le \frac{a_i}{b_{i+1}} \le \max\left(\frac{a_i}{b_{i+1}}, \frac{1}{b_i}\right)$$
- 若都是第二项大，则 $b_i = b_{i+1}$，则 $a_i = a_{i+1} = 1$，之前的结论仍然可用

NOIP2012 国王游戏

NOIP2012 国王游戏

- 假设只是原式第二项大，由上式可知交换不会优，同时由于有 $\frac{1}{b_i} \geq \frac{a_i}{b_{i+1}}$，可得 $b_{i+1} \geq b_i \cdot a_i$，即 $a_i \cdot b_i \leq a_{i+1} \cdot b_{i+1}$

NOIP2012 国王游戏

- 假设只是原式第二项大，由上式可知交换不会优，同时由于有 $\frac{1}{b_i} \geq \frac{a_i}{b_{i+1}}$，可得 $b_{i+1} \geq b_i \cdot a_i$，即 $a_i \cdot b_i \leq a_{i+1} \cdot b_{i+1}$
- 这个式子就是之前的结论

NOIP2012 国王游戏

- 假设只是原式第二项大，由上式可知交换不会优，同时由于有 $\frac{1}{b_i} \geq \frac{a_i}{b_{i+1}}$，可得 $b_{i+1} \geq b_i \cdot a_i$，即 $a_i \cdot b_i \leq a_{i+1} \cdot b_{i+1}$
- 这个式子就是之前的结论
- 如果只是新解第二项大，我们可以类似上面的推导得到同样的式子

NOIP2012 国王游戏

- 假设只是原式第二项大，由上式可知交换不会优，同时由于有 $\frac{1}{b_i} \geq \frac{a_i}{b_{i+1}}$，可得 $b_{i+1} \geq b_i \cdot a_i$，即 $a_i \cdot b_i \leq a_{i+1} \cdot b_{i+1}$
- 这个式子就是之前的结论
- 如果只是新解第二项大，我们可以类似上面的推导得到同样的式子
- 这个式子满足传递性，所有组按 $a_i \cdot b_i$ 排序即可

NOIP2012 国王游戏

- 假设只是原式第二项大，由上式可知交换不会优，同时由于有 $\frac{1}{b_i} \geq \frac{a_i}{b_{i+1}}$，可得 $b_{i+1} \geq b_i \cdot a_i$，即 $a_i \cdot b_i \leq a_{i+1} \cdot b_{i+1}$
- 这个式子就是之前的结论
- 如果只是新解第二项大，我们可以类似上面的推导得到同样的式子
- 这个式子满足传递性，所有组按 $a_i \cdot b_i$ 排序即可
- 放宽条件或强制约束并逐步分析，得出最终结论

Part II

技巧选讲

Outline

坐标离散化

坐标离散化

- $w \times h$ 的格子上画了 n 条垂直或是水平的宽度为 1 长度为 l_i 的直线，求出这些线将格子划分成了多少个区域

坐标离散化

- $w \times h$ 的格子上画了 n 条垂直或是水平的宽度为 1 长度为 l_i 的直线，求出这些线将格子划分成了多少个区域
- $1 \leq w, h \leq 10^6$, $1 \leq n \leq 500$

坐标离散化

- $w \times h$ 的格子上画了 n 条垂直或是水平的宽度为 1 长度为 l_i 的直线，求出这些线将格子划分成了多少个区域
- $1 \le w, h \le 10^6$, $1 \le n \le 500$
- 求分成了多少区域可以使用 floodfill，然而坐标范围太大，因此需要用坐标离散化这一技巧

坐标离散化

- $w \times h$ 的格子上画了 n 条垂直或是水平的宽度为 1 长度为 l_i 的直线，求出这些线将格子划分成了多少个区域
- $1 \leq w, h \leq 10^6$, $1 \leq n \leq 500$
- 求分成了多少区域可以使用 floodfill，然而坐标范围太大，因此需要用坐标离散化这一技巧
- 坐标离散化即只保留对计算有用的坐标

坐标离散化

- $w \times h$ 的格子上画了 n 条垂直或是水平的宽度为 1 长度为 l_i 的直线，求出这些线将格子划分成了多少个区域
- $1 \le w, h \le 10^6$, $1 \le n \le 500$
- 求分成了多少区域可以使用 floodfill，然而坐标范围太大，因此需要用坐标离散化这一技巧
- 坐标离散化即只保留对计算有用的坐标
- 消除对答案没有影响的行列

坐标离散化

- $w \times h$ 的格子上画了 n 条垂直或是水平的宽度为 1 长度为 l_i 的直线，求出这些线将格子划分成了多少个区域
- $1 \le w, h \le 10^6$, $1 \le n \le 500$
- 求分成了多少区域可以使用 floodfill，然而坐标范围太大，因此需要用坐标离散化这一技巧
- 坐标离散化即只保留对计算有用的坐标
- 消除对答案没有影响的行列
- 本题中只要保存有直线的行列以及它前后的行列即可，坐标范围最多 $6n \times 6n$

坐标离散化

坐标离散化

- 还可以应用在一些几何题以及一些序列问题

坐标离散化

- 还可以应用在一些几何题以及一些序列问题
- 序列有 10^9 个数，初始都为 0，现在有 $m(m \leq 10^5)$ 个操作，每个操作影响一个区间

坐标离散化

- 还可以应用在一些几何题以及一些序列问题
- 序列有 10^9 个数，初始都为 0，现在有 $m(m \leq 10^5)$ 个操作，每个操作影响一个区间
- 只需要存下每个区间的左右端点 l_i, r_i，其他位置都统一处理

坐标离散化

- 还可以应用在一些几何题以及一些序列问题
- 序列有 10^9 个数，初始都为 0，现在有 $m(m \le 10^5)$ 个操作，每个操作影响一个区间
- 只需要存下每个区间的左右端点 l_i, r_i，其他位置都统一处理
- 给定 n 个矩形，求它们的面积并。$n \le 100$, $|坐标范围| \le 10^9$

坐标离散化

- 还可以应用在一些几何题以及一些序列问题
- 序列有 10^9 个数，初始都为 0，现在有 $m(m \le 10^5)$ 个操作，每个操作影响一个区间
- 只需要存下每个区间的左右端点 l_i, r_i，其他位置都统一处理
- 给定 n 个矩形，求它们的面积并。$n \le 100$, $|坐标范围| \le 10^9$
- 存下矩形四个端点的坐标值，其它坐标统一处理

坐标离散化

- 还可以应用在一些几何题以及一些序列问题
- 序列有 10^9 个数，初始都为 0，现在有 $m(m \le 10^5)$ 个操作，每个操作影响一个区间
- 只需要存下每个区间的左右端点 l_i, r_i，其他位置都统一处理
- 给定 n 个矩形，求它们的面积并。$n \le 100$, $|坐标范围| \le 10^9$
- 存下矩形四个端点的坐标值，其它坐标统一处理
- 通常需要求出两个新坐标间的长度

Outline

前缀和

前缀和

- 给定长为 n 的序列 a_i，m 次询问区间 $[l_i, r_i]$ 内数的和，$n, m \leq 10^5$

前缀和

- 给定长度为 n 的序列 a_i，m 次询问区间 $[l_i, r_i]$ 内数的和，$n, m \leq 10^5$
- 令 $sum_i = \sum_{j=1}^{i} a_j$，则一次询问答案为 $sum_r - sum_{l-1}$

前缀和

- 给定长为 n 的序列 a_i，m 次询问区间 $[l_i, r_i]$ 内数的和，$n, m \le 10^5$
- 令 $sum_i = \sum_{j=1}^{i} a_j$，则一次询问答案为 $sum_r - sum_{l-1}$
- m 次询问，求 $[l, r]$ 内的整数一共含有多少个质数 p 的因子，$m \le 10^5$，$l, r, p \le 10^9$

前缀和

- 给定长为 n 的序列 a_i，m 次询问区间 $[l_i, r_i]$ 内数的和，$n, m \leq 10^5$
- 令 $sum_i = \sum_{j=1}^{i} a_j$，则一次询问答案为 $sum_r - sum_{l-1}$
- m 次询问，求 $[l, r]$ 内的整数一共含有多少个质数 p 的因子，$m \leq 10^5$，$l, r, p \leq 10^9$
- 设 sum_i 表示 $[1, i]$ 内的整数一共含有多少个质数 p 的因子，答案为 $sum_r - sum_{l-1}$

前缀和

- 给定长为 n 的序列 a_i，m 次询问区间 $[l_i, r_i]$ 内数的和，$n, m \leq 10^5$
- 令 $sum_i = \sum_{j=1}^{i} a_j$，则一次询问答案为 $sum_r - sum_{l-1}$
- m 次询问，求 $[l, r]$ 内的整数一共含有多少个质数 p 的因子，$m \leq 10^5$，$l, r, p \leq 10^9$
- 设 sum_i 表示 $[1, i]$ 内的整数一共含有多少个质数 p 的因子，答案为 $sum_r - sum_{l-1}$
- $sum_n = \sum_{i=1}^{\infty} \lfloor \frac{n}{p^i} \rfloor$

前缀和

- 给定长为 n 的序列 a_i，m 次询问区间 $[l_i, r_i]$ 内数的和，$n, m \le 10^5$
- 令 $sum_i = \sum_{j=1}^{i} a_j$，则一次询问答案为 $sum_r - sum_{l-1}$
- m 次询问，求 $[l, r]$ 内的整数一共含有多少个质数 p 的因子，$m \le 10^5$，$l, r, p \le 10^9$
- 设 sum_i 表示 $[1, i]$ 内的整数一共含有多少个质数 p 的因子，答案为 $sum_r - sum_{l-1}$
- $sum_n = \sum_{i=1}^{\infty} \lfloor \frac{n}{p^i} \rfloor$
- 给定 n 个数 a_i, b_i，对每个 i 求 $b_i \cdot \max\limits_{1 \le j \le n 且 j \ne i} a_j$

前缀和

- 给定长为 n 的序列 a_i，m 次询问区间 $[l_i, r_i]$ 内数的和，$n, m \leq 10^5$
- 令 $sum_i = \sum_{j=1}^{i} a_j$，则一次询问答案为 $sum_r - sum_{l-1}$
- m 次询问，求 $[l, r]$ 内的整数一共含有多少个质数 p 的因子，$m \leq 10^5$，$l, r, p \leq 10^9$
- 设 sum_i 表示 $[1, i]$ 内的整数一共含有多少个质数 p 的因子，答案为 $sum_r - sum_{l-1}$
- $sum_n = \sum_{i=1}^{\infty} \lfloor \frac{n}{p^i} \rfloor$
- 给定 n 个数 a_i, b_i，对每个 i 求 $b_i \cdot \max_{1 \leq j \leq n 且 j \neq i} a_j$
- 求一个前缀最大值，再求一个后缀最大值，每个位置 $O(1)$ 计算答案

前缀和

前缀和

- 经常配合差分使用

前缀和

- 经常配合差分使用
- 一个 $n \times m$ 的网格，每个网格里数字初始为 0，现在有 q 次操作，每次操作对一个矩形整体加一个数，或是询问某个单点格子的数 $a_{x,y}$ 是多少

前缀和

- 经常配合差分使用
- 一个 $n \times m$ 的网格，每个网格里数字初始为 0，现在有 q 次操作，每次操作对一个矩形整体加一个数，或是询问某个单点格子的数 $a_{x,y}$ 是多少
- $n, m \leq 1000, q \leq 10^5$

前缀和

- 经常配合差分使用
- 一个 $n \times m$ 的网格，每个网格里数字初始为 0，现在有 q 次操作，每次操作对一个矩形整体加一个数，或是询问某个单点格子的数 $a_{x,y}$ 是多少
- $n, m \leq 1000, q \leq 10^5$
- 对一个左上角为 $[x_1, y_1]$ 右下角为 $[x_2, y_2]$ 的矩形整体加数 d，我们可以把它差分为 $s[x_2, y_2] + d, s[x_2, y_1 - 1] - d, s[x_1 - 1, y_2] - d, s[x_1 - 1, y_1 - 1] + d$，这样 $a_{x,y}$ 的值就为 $[1, 1] \sim [x, y]$ 这个矩阵 s 的和

前缀和

- 经常配合差分使用
- 一个 $n \times m$ 的网格，每个网格里数字初始为 0，现在有 q 次操作，每次操作对一个矩形整体加一个数，或是询问某个单点格子的数 $a_{x,y}$ 是多少
- $n, m \leq 1000, q \leq 10^5$
- 对一个左上角为 $[x_1, y_1]$ 右下角为 $[x_2, y_2]$ 的矩形整体加数 d，我们可以把它差分为 $s[x_2, y_2] + d, s[x_2, y_1 - 1] - d, s[x_1 - 1, y_2] - d, s[x_1 - 1, y_1 - 1] + d$，这样 $a_{x,y}$ 的值就为 $[1, 1] \sim [x, y]$ 这个矩阵 s 的和
- 若是单点修改矩形询问，则把询问做二维前缀和即可

前缀和

- 经常配合差分使用
- 一个 $n \times m$ 的网格，每个网格里数字初始为 0，现在有 q 次操作，每次操作对一个矩形整体加一个数，或是询问某个单点格子的数 $a_{x,y}$ 是多少
- $n, m \leq 1000, q \leq 10^5$
- 对一个左上角为 $[x_1, y_1]$ 右下角为 $[x_2, y_2]$ 的矩形整体加数 d，我们可以把它差分为 $s[x_2, y_2] + d, s[x_2, y_1 - 1] - d, s[x_1 - 1, y_2] - d, s[x_1 - 1, y_1 - 1] + d$，这样 $a_{x,y}$ 的值就为 $[1, 1] \sim [x, y]$ 这个矩阵 s 的和
- 若是单点修改矩形询问，则把询问做二维前缀和即可
- $sum_{x,y} = \sum_{i=1}^{x} \sum_{j=1}^{y} a_{x,y}$

前缀和

- 经常配合差分使用
- 一个 $n \times m$ 的网格，每个网格里数字初始为 0，现在有 q 次操作，每次操作对一个矩形整体加一个数，或是询问某个单点格子的数 $a_{x,y}$ 是多少
- $n, m \le 1000, q \le 10^5$
- 对一个左上角为 $[x_1, y_1]$ 右下角为 $[x_2, y_2]$ 的矩形整体加数 d，我们可以把它差分为 $s[x_2, y_2] + d, s[x_2, y_1 - 1] - d, s[x_1 - 1, y_2] - d, s[x_1 - 1, y_1 - 1] + d$，这样 $a_{x,y}$ 的值就为 $[1, 1] \sim [x, y]$ 这个矩阵 s 的和
- 若是单点修改矩形询问，则把询问做二维前缀和即可
- $sum_{x,y} = \sum_{i=1}^{x} \sum_{j=1}^{y} a_{x,y}$
- $ans = sum_{x_2,y_2} - sum_{x_2,y_1-1} - sum_{x_1-1,y_2}, + sum_{x_1-1,y_1-1}$

Outline

绝对值最大

绝对值最大

- 给定 d 维空间中 n 个人的坐标，定义距离为曼哈顿距离。求离的最远的两个人距离是多少

绝对值最大

- 给定 d 维空间中 n 个人的坐标，定义距离为曼哈顿距离。求离的最远的两个人距离是多少
- $n \leq 10^4$，$d \leq 10$

绝对值最大

- 给定 d 维空间中 n 个人的坐标，定义距离为曼哈顿距离。求离的最远的两个人距离是多少
- $n \leq 10^4$, $d \leq 10$
- 方便起见，先考虑二维的情况

绝对值最大

- 给定 d 维空间中 n 个人的坐标，定义距离为曼哈顿距离。求离的最远的两个人距离是多少
- $n \le 10^4$, $d \le 10$
- 方便起见，先考虑二维的情况
- 考虑第 i 个人，离他最远的人的距离

绝对值最大

- 给定 d 维空间中 n 个人的坐标，定义距离为曼哈顿距离。求离的最远的两个人距离是多少
- $n \leq 10^4$ ，$d \leq 10$
- 方便起见，先考虑二维的情况
- 考虑第 i 个人，离他最远的人的距离
- 根据第 i 个人的坐标，把坐标轴切成四个小矩形，即当前点的左上右上左下右下区域

绝对值最大

- 给定 d 维空间中 n 个人的坐标，定义距离为曼哈顿距离。求离的最远的两个人距离是多少
- $n \leq 10^4$ ，$d \leq 10$
- 方便起见，先考虑二维的情况
- 考虑第 i 个人，离他最远的人的距离
- 根据第 i 个人的坐标，把坐标轴切成四个小矩形，即当前点的左上右上左下右下区域
- 每个区域内曼哈顿距离中的绝对值符号已知

绝对值最大

- 给定 d 维空间中 n 个人的坐标，定义距离为曼哈顿距离。求离的最远的两个人距离是多少
- $n \leq 10^4$ ，$d \leq 10$
- 方便起见，先考虑二维的情况
- 考虑第 i 个人，离他最远的人的距离
- 根据第 i 个人的坐标，把坐标轴切成四个小矩形，即当前点的左上右上左下右下区域
- 每个区域内曼哈顿距离中的绝对值符号已知
- 如右上角，$|x_i - x_j| + |y_i - y_j| = x_j - x_i + y_i - y_j = (-x_i + y_i) - (-x_j + y_j)$

绝对值最大

- 给定 d 维空间中 n 个人的坐标，定义距离为曼哈顿距离。求离的最远的两个人距离是多少
- $n \leq 10^4$ ，$d \leq 10$
- 方便起见，先考虑二维的情况
- 考虑第 i 个人，离他最远的人的距离
- 根据第 i 个人的坐标，把坐标轴切成四个小矩形，即当前点的左上右上左下右下区域
- 每个区域内曼哈顿距离中的绝对值符号已知
- 如右上角，$|x_i - x_j| + |y_i - y_j| = x_j - x_i + y_i - y_j = (-x_i + y_i) - (-x_j + y_j)$
- 每个角都可以将距离划为只与一个点有关的形式

绝对值最大

绝对值最大

- 由于绝对值只有在符号正确时才取得最大值，因此不必知道每个点真正在当前点哪个角

绝对值最大

- 由于绝对值只有在符号正确时才取得最大值，因此不必知道每个点真正在当前点哪个角
- 每个点计算 2^d 种情况，用于更新对应情况坐标和的最大最小值

绝对值最大

- 由于绝对值只有在符号正确时才取得最大值，因此不必知道每个点真正在当前点哪个角
- 每个点计算 2^d 种情况，用于更新对应情况坐标和的最大最小值
- 每种情况最大最小值相减后即为这种情况下最远的两个人的距离

绝对值最大

- 由于绝对值只有在符号正确时才取得最大值，因此不必知道每个点真正在当前点哪个角
- 每个点计算 2^d 种情况，用于更新对应情况坐标和的最大最小值
- 每种情况最大最小值相减后即为这种情况下最远的两个人的距离
- 若某种情况求出的值不是真实情况也没有关系，因为真实情况的那个值一定更大，所有情况取最大值后不会影响答案正确性

绝对值最大

- 由于绝对值只有在符号正确时才取得最大值，因此不必知道每个点真正在当前点哪个角
- 每个点计算 2^d 种情况，用于更新对应情况坐标和的最大最小值
- 每种情况最大最小值相减后即为这种情况下最远的两个人的距离
- 若某种情况求出的值不是真实情况也没有关系，因为真实情况的那个值一定更大，所有情况取最大值后不会影响答案正确性
- 一个 $n \times m$ 的网格，每个格子上有两个属性 $A_{i,j}, B_{i,j}$，两个格子之间的距离为曼哈顿距离，求一条路径，使得路径上的格子的 $A_{i,j}$ 严格递增，且最大化总移动距离加上路径中所有格子 $B_{i,j}$ 的和，$1 \le n, m \le 1000$，$1 \le A_{i,j}, B_{i,j} \le 10^6$

绝对值最大

- 由于绝对值只有在符号正确时才取得最大值，因此不必知道每个点真正在当前点哪个角
- 每个点计算 2^d 种情况，用于更新对应情况坐标和的最大最小值
- 每种情况最大最小值相减后即为这种情况下最远的两个人的距离
- 若某种情况求出的值不是真实情况也没有关系，因为真实情况的那个值一定更大，所有情况取最大值后不会影响答案正确性
- 一个 $n \times m$ 的网格，每个格子上有两个属性 $A_{i,j}, B_{i,j}$，两个格子之间的距离为曼哈顿距离，求一条路径，使得路径上的格子的 $A_{i,j}$ 严格递增，且最大化总移动距离加上路径中所有格子 $B_{i,j}$ 的和，$1 \le n, m \le 1000$，$1 \le A_{i,j}, B_{i,j} \le 10^6$
- 排序后考虑 $f_{i,j}$ 表示以 (i,j) 为结尾的最长路径

绝对值最大

- 由于绝对值只有在符号正确时才取得最大值，因此不必知道每个点真正在当前点哪个角
- 每个点计算 2^d 种情况，用于更新对应情况坐标和的最大最小值
- 每种情况最大最小值相减后即为这种情况下最远的两个人的距离
- 若某种情况求出的值不是真实情况也没有关系，因为真实情况的那个值一定更大，所有情况取最大值后不会影响答案正确性
- 一个 $n \times m$ 的网格，每个格子上有两个属性 $A_{i,j}, B_{i,j}$，两个格子之间的距离为曼哈顿距离，求一条路径，使得路径上的格子的 $A_{i,j}$ 严格递增，且最大化总移动距离加上路径中所有格子 $B_{i,j}$ 的和，$1 \le n, m \le 1000$，$1 \le A_{i,j}, B_{i,j} \le 10^6$
- 排序后考虑 $f_{i,j}$ 表示以 (i,j) 为结尾的最长路径
- 依然将曼哈顿距离分成四种情况讨论，$f_{i,j}$ 用最大的转移即可

Outline

逆向思维

逆向思维

- 给 n 个数字 a_i，现在每次操作可以任意更改某个数字为任意值，求最少几次操作使得 a_i 不降，$n \le 10^5$

逆向思维

- 给 n 个数字 a_i，现在每次操作可以任意更改某个数字为任意值，求最少几次操作使得 a_i 不降，$n \le 10^5$
- 修改数字最少 $\Rightarrow$ 不修改 (保留) 的数字最多 $\Rightarrow$ 求原序列 LIS

逆向思维

- 给 n 个数字 a_i，现在每次操作可以任意更改某个数字为任意值，求最少几次操作使得 a_i 不降，$n \leq 10^5$
- 修改数字最少 $\Rightarrow$ 不修改 (保留) 的数字最多 $\Rightarrow$ 求原序列 LIS
- 对所求答案换个角度思考，如删除数字 $\Rightarrow$ 保留数字

逆向思维

- 给 n 个数字 a_i，现在每次操作可以任意更改某个数字为任意值，求最少几次操作使得 a_i 不降，$n \leq 10^5$
- 修改数字最少 $\Rightarrow$ 不修改 (保留) 的数字最多 $\Rightarrow$ 求原序列 LIS
- 对所求答案换个角度思考，如删除数字 $\Rightarrow$ 保留数字
- 给一张 n 个点 m 条边的无向图，q 次操作，每次操作删一条边，问每次删除后图里还有几个联通块，$n, m, q \leq 10^5$

逆向思维

- 给 n 个数字 a_i，现在每次操作可以任意更改某个数字为任意值，求最少几次操作使得 a_i 不降，$n \le 10^5$
- 修改数字最少 $\Rightarrow$ 不修改 (保留) 的数字最多 $\Rightarrow$ 求原序列 LIS
- 对所求答案换个角度思考，如删除数字 $\Rightarrow$ 保留数字
- 给一张 n 个点 m 条边的无向图，q 次操作，每次操作删一条边，问每次删除后图里还有几个联通块，$n, m, q \le 10^5$
- 倒着考虑操作，变为加边，并查集维护

逆向思维

- 给 n 个数字 a_i，现在每次操作可以任意更改某个数字为任意值，求最少几次操作使得 a_i 不降，$n \le 10^5$
- 修改数字最少 $\Rightarrow$ 不修改 (保留) 的数字最多 $\Rightarrow$ 求原序列 LIS
- 对所求答案换个角度思考，如删除数字 $\Rightarrow$ 保留数字
- 给一张 n 个点 m 条边的无向图，q 次操作，每次操作删一条边，问每次删除后图里还有几个联通块，$n, m, q \le 10^5$
- 倒着考虑操作，变为加边，并查集维护
- 对于一类可离线的涉及删除 (插入) 的操作问题，可以倒着考虑操作，将操作变为插入 (删除)

逆向思维

逆向思维

- 给定一个 n 位十进制数 L，L_i 表示第 i 位数字大小 (不含前导 0)，求有多少个不超过 n 位的非负十进制数满足至少有一个数位数字大小不小于 L 对应的数字，不满 n 位高位补 0. 答案模大质数

逆向思维

- 给定一个 n 位十进制数 L，L_i 表示第 i 位数字大小 (不含前导 0)，求有多少个不超过 n 位的非负十进制数满足至少有一个数位数字大小不小于 L 对应的数字，不满 n 位高位补 0. 答案模大质数
- 至少一位不小于 $\Rightarrow$ 总数减去每一位都小于，$ans = 10^n - \prod_{i=1}^{n} L_i$

逆向思维

- 给定一个 n 位十进制数 L，L_i 表示第 i 位数字大小 (不含前导 0)，求有多少个不超过 n 位的非负十进制数满足至少有一个数位数字大小不小于 L 对应的数字，不满 n 位高位补 0. 答案模大质数
- 至少一位不小于 $\Rightarrow$ 总数减去每一位都小于，$ans = 10^n - \prod_{i=1}^{n} L_i$
- 求满足某个条件的元素个数时，若直接求不好求，可反过来考虑求不满足条件的元素个数，用总数减去即是答案

逆向思维

- 给定一个 n 位十进制数 L，L_i 表示第 i 位数字大小 (不含前导 0)，求有多少个不超过 n 位的非负十进制数满足至少有一个数位数字大小不小于 L 对应的数字，不满 n 位高位补 0. 答案模大质数
- 至少一位不小于 $\Rightarrow$ 总数减去每一位都小于，$ans = 10^n - \prod_{i=1}^{n} L_i$
- 求满足某个条件的元素个数时，若直接求不好求，可反过来考虑求不满足条件的元素个数，用总数减去即是答案
- 容斥原理的应用

Outline

容斥原理

容斥原理

- $$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i=1}^{n} |A_i| - \sum_{1\le i<j\le n} \left|A_i \bigcap A_j\right| + \sum_{1\le i<j<k\le n} \left|A_i \bigcap A_j \bigcap A_k\right| - \cdots + (-1)^{n-1} \left|A_1 \bigcap \cdots \cap A_n\right|$$

容斥原理

- $$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i=1}^{n} |A_i| - \sum_{1\le i<j\le n} \left|A_i \bigcap A_j\right| + \sum_{1\le i<j<k\le n} \left|A_i \bigcap A_j \bigcap A_k\right| - \cdots + (-1)^{n-1}\left|A_1 \bigcap \cdots \cap A_n\right|$$

- $$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{\varnothing\neq J\subseteq\{1,2,\cdots,n\}} (-1)^{|J|-1}\left|\bigcap_{j\in J} A_j\right|$$

容斥原理

- $$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i=1}^{n} |A_i| - \sum_{1\le i<j\le n} \left|A_i \bigcap A_j\right| + \sum_{1\le i<j<k\le n} \left|A_i \bigcap A_j \bigcap A_k\right| - \cdots + (-1)^{n-1} \left|A_1 \bigcap \cdots \cap A_n\right|$$

- $$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{\varnothing\neq J\subseteq\{1,2,\cdots,n\}} (-1)^{|J|-1} \left|\bigcap_{j\in J} A_j\right|$$

- $$\left|\bigcap_{i=1}^{n} \bar{A}_i\right| = \left|S - \bigcup_{i=1}^{n} A_i\right| = |S| - \left|\bigcup_{i=1}^{n} A_i\right|$$

容斥原理

容斥原理

- Special case :

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{k=1}^{n}(-1)^{k-1}\binom{n}{k}a_k$$

容斥原理

- Special case :

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{k=1}^{n}(-1)^{k-1}\binom{n}{k}a_k$$

- Special case :

$$\left|\bigcap_{i=1}^{n} \bar{A}_i\right| = \sum_{k=0}^{n}(-1)^{k}\binom{n}{k}a_k$$

容斥原理

- Special case :

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{k=1}^{n}(-1)^{k-1}\binom{n}{k}a_k$$

- Special case :

$$\left|\bigcap_{i=1}^{n} \bar{A}_i\right| = \sum_{k=0}^{n}(-1)^{k}\binom{n}{k}a_k$$

-

$$f(n) = \sum_{d|n} g(d) \Longleftrightarrow g(n) = \sum_{d|n} \mu(\frac{n}{d})f(d)$$

容斥原理

- Special case：

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{k=1}^{n}(-1)^{k-1}\binom{n}{k}a_k$$

- Special case：

$$\left|\bigcap_{i=1}^{n} \bar{A}_i\right| = \sum_{k=0}^{n}(-1)^{k}\binom{n}{k}a_k$$

-

$$f(n) = \sum_{d|n} g(d) \Longleftrightarrow g(n) = \sum_{d|n} \mu(\frac{n}{d}) f(d)$$

-

$$f(n) = \sum_{n|d} g(d) \Longleftrightarrow g(n) = \sum_{n|d} \mu(\frac{d}{n}) f(d)$$

容斥原理

容斥原理

- $$\mu(n)=\begin{cases}1 & n=1\\ 0 & \exists d>1, d^2 \mid n\\ (-1)^k & n=\prod_{i=1}^{k} p_i\end{cases}$$

容斥原理

- $$\mu(n)=\begin{cases}1 & n=1\\ 0 & \exists d>1, d^2 \mid n\\ (-1)^k & n=\prod_{i=1}^{k} p_i\end{cases}$$
- 求所有由小写字母组成的长度为 n 的字符串中，最小循环节长度为 n 的字符串个数，答案模大质数，$2 \le n \le 10^9$

容斥原理

- $$\mu(n) = \begin{cases} 1 & n = 1 \\ 0 & \exists d > 1, d^2 \mid n \\ (-1)^k & n = \prod_{i=1}^{k} p_i \end{cases}$$
- 求所有由小写字母组成的长度为 n 的字符串中，最小循环节长度为 n 的字符串个数，答案模大质数，$2 \le n \le 10^9$
- 令最小循环节长度恰好为 d 的字符串个数为 $g(d)$，最小循环节长度为 d 的约数的字符串个数为 $f(d)$

容斥原理

- $$\mu(n)=\begin{cases}1 & n=1\\ 0 & \exists d>1, d^2 \mid n\\ (-1)^k & n=\prod_{i=1}^{k} p_i\end{cases}$$
- 求所有由小写字母组成的长度为 n 的字符串中，最小循环节长度为 n 的字符串个数，答案模大质数，$2\le n\le 10^9$
- 令最小循环节长度恰好为 d 的字符串个数为 $g(d)$，最小循环节长度为 d 的约数的字符串个数为 $f(d)$
- $f(n)=\sum_{d|n} g(d)$ ，$g(n)=\sum_{d|n}\mu(\frac{n}{d})f(d)$

容斥原理

- $$\mu(n)=\begin{cases}1 & n=1\\ 0 & \exists d>1, d^2 \mid n\\ (-1)^k & n=\prod_{i=1}^{k} p_i\end{cases}$$
- 求所有由小写字母组成的长度为 n 的字符串中，最小循环节长度为 n 的字符串个数，答案模大质数，$2 \le n \le 10^9$
- 令最小循环节长度恰好为 d 的字符串个数为 $g(d)$，最小循环节长度为 d 的约数的字符串个数为 $f(d)$
- $f(n)=\sum_{d|n} g(d)$, $g(n)=\sum_{d|n} \mu(\frac{n}{d}) f(d)$
- $f(n)=26^n$

容斥原理

- $$\mu(n)=\begin{cases}1 & n=1\\ 0 & \exists d>1, d^2 \mid n\\ (-1)^k & n=\prod_{i=1}^{k} p_i\end{cases}$$
- 求所有由小写字母组成的长度为 n 的字符串中，最小循环节长度为 n 的字符串个数，答案模大质数，$2 \leq n \leq 10^9$
- 令最小循环节长度恰好为 d 的字符串个数为 $g(d)$，最小循环节长度为 d 的约数的字符串个数为 $f(d)$
- $f(n)=\sum_{d|n} g(d)$, $g(n)=\sum_{d|n} \mu(\frac{n}{d})f(d)$
- $f(n)=26^n$
- DFS 出 n 的所有约数并算出 mu 与 f 的值，利用反演得出答案

容斥原理

容斥原理

- 给定一个数集 S，求 $[1, n]$ 内有多少个数至少是 S 中一个数的倍数，$|S| \leq 15, n \leq 10^{18}$

容斥原理

- 给定一个数集 S，求 $[1,n]$ 内有多少个数至少是 S 中一个数的倍数，$|S| \le 15, n \le 10^{18}$
- 容斥原理直接套用，A_i 表示 $[1,n]$ 中是 S_i 倍数的数的个数，$A_i = \lfloor \frac{n}{S_i} \rfloor$

容斥原理

- 给定一个数集 S，求 $[1, n]$ 内有多少个数至少是 S 中一个数的倍数，$|S| \leq 15, n \leq 10^{18}$
- 容斥原理直接套用，A_i 表示 $[1, n]$ 中是 S_i 倍数的数的个数，$A_i = \lfloor \frac{n}{S_i} \rfloor$
- $\left| A_i \cap A_j \right| = \lfloor \frac{n}{\mathrm{lcm}(S_i, S_j)} \rfloor$

容斥原理

- 给定一个数集 S，求 $[1,n]$ 内有多少个数至少是 S 中一个数的倍数，$|S| \leq 15, n \leq 10^{18}$
- 容斥原理直接套用，A_i 表示 $[1,n]$ 中是 S_i 倍数的数的个数，$A_i = \lfloor \frac{n}{S_i} \rfloor$
- $\left|A_i \cap A_j\right| = \lfloor \frac{n}{\mathrm{lcm}(S_i,S_j)} \rfloor$
- 求有多少个长度为 n 的排列 p 满足 $\exists i, p_i = i$

容斥原理

- 给定一个数集 S，求 $[1,n]$ 内有多少个数至少是 S 中一个数的倍数，$|S| \le 15, n \le 10^{18}$
- 容斥原理直接套用，A_i 表示 $[1,n]$ 中是 S_i 倍数的数的个数，$A_i = \lfloor \frac{n}{S_i} \rfloor$
- $\left|A_i \cap A_j\right| = \lfloor \frac{n}{\operatorname{lcm}(S_i,S_j)} \rfloor$
- 求有多少个长度为 n 的排列 p 满足 $\exists i, p_i = i$
- 设 A_i 表示满足 $p_i = i$ 的排列数，$A_i = (n-1)!$

容斥原理

- 给定一个数集 S，求 $[1,n]$ 内有多少个数至少是 S 中一个数的倍数，$|S| \le 15, n \le 10^{18}$
- 容斥原理直接套用，A_i 表示 $[1,n]$ 中是 S_i 倍数的数的个数，$A_i = \lfloor \frac{n}{S_i} \rfloor$
- $\left|A_i \cap A_j\right| = \lfloor \frac{n}{\mathrm{lcm}(S_i,S_j)} \rfloor$
- 求有多少个长度为 n 的排列 p 满足 $\exists i, p_i = i$
- 设 A_i 表示满足 $p_i = i$ 的排列数，$A_i = (n-1)!$
- 容斥原理的特殊形式

容斥原理

- 给定一个数集 S，求 $[1, n]$ 内有多少个数至少是 S 中一个数的倍数，$|S| \le 15, n \le 10^{18}$
- 容斥原理直接套用，A_i 表示 $[1, n]$ 中是 S_i 倍数的数的个数，$A_i = \lfloor \frac{n}{S_i} \rfloor$
- $\left| A_i \cap A_j \right| = \lfloor \frac{n}{\operatorname{lcm}(S_i, S_j)} \rfloor$
- 求有多少个长度为 n 的排列 p 满足 $\exists i, p_i = i$
- 设 A_i 表示满足 $p_i = i$ 的排列数，$A_i = (n-1)!$
- 容斥原理的特殊形式
- $ans = \sum\limits_{k=1}^{n} (-1)^{k-1} \binom{n}{k} (n-k)!$

Outline

其他小技巧

其他小技巧

- $\lim\limits_{k\to\infty}\sum\limits_{i=1}^{k}\frac{n}{i} = O(n\log n)$

其他小技巧

- $\lim_{k\to\infty}\sum_{i=1}^{k}\frac{n}{i}=O(n\log n)$
- 枚举平均分成的段数暴力的复杂度保证

其他小技巧

- $\lim\limits_{k\to\infty}\sum\limits_{i=1}^{k}\frac{n}{i}=O(n\log n)$
- 枚举平均分成的段数暴力的复杂度保证
- 抽屉原理

其他小技巧

- $\lim\limits_{k\to\infty}\sum\limits_{i=1}^{k}\frac{n}{i} = O(n\log n)$
- 枚举平均分成的段数暴力的复杂度保证
- 抽屉原理
- 给定数列 a_i，问区间 $[l, r]$ 内和模 p 为 0 的子串是否存在，$p \leq 100$

其他小技巧

- $\lim\limits_{k\to\infty}\sum\limits_{i=1}^{k}\frac{n}{i}=O(n\log n)$
- 枚举平均分成的段数暴力的复杂度保证
- 抽屉原理
- 给定数列 a_i，问区间 $[l,r]$ 内和模 p 为 0 的子串是否存在，$p\le 100$
- $r-l+1\ge p$ 必定存在，否则暴力

其他小技巧

- $\lim\limits_{k\to\infty}\sum\limits_{i=1}^{k}\frac{n}{i}=O(n\log n)$
- 枚举平均分成的段数暴力的复杂度保证
- 抽屉原理
- 给定数列 a_i，问区间 $[l,r]$ 内和模 p 为 0 的子串是否存在，$p\le 100$
- $r-l+1\ge p$ 必定存在，否则暴力
- min-max 容斥

其他小技巧

- $\lim_{k\to\infty}\sum_{i=1}^{k}\frac{n}{i}=O(n\log n)$
- 枚举平均分成的段数暴力的复杂度保证
- 抽屉原理
- 给定数列 a_i，问区间 $[l,r]$ 内和模 p 为 0 的子串是否存在，$p\leq 100$
- $r-l+1\geq p$ 必定存在，否则暴力
- min-max 容斥
- $\max(S)=\sum_{s\subseteq S}(-1)^{|s|+1}\min(s)$

其他小技巧

- $\lim_{k\to\infty} \sum_{i=1}^{k} \frac{n}{i} = O(n \log n)$
- 枚举平均分成的段数暴力的复杂度保证
- 抽屉原理
- 给定数列 a_i，问区间 $[l, r]$ 内和模 p 为 0 的子串是否存在，$p \le 100$
- $r - l + 1 \ge p$ 必定存在，否则暴力
- min-max 容斥
- $\max(S) = \sum_{s \subseteq S} (-1)^{|s|+1} \min(s)$
- $\mathrm{lcm}(S) = \prod_{s \subseteq S} (\gcd(s))^{(-1)^{|s|+1}}$，这里 a^{-1} 表示除以 a

Part III

Ending

参考资料

[1] 秋叶拓哉，岩田阳一，北川宜稔，挑战程序设计竞赛 (第 2 版)，巫泽俊译

[2] 刘汝佳，黄亮，算法艺术与信息学竞赛

[3] 维基百科，容斥原理

Thanks for listening

QQ:1641901772

欢迎提问