第六周作业讲评

LeavingZ

2025/3/8

www.luogu.com.cn

目录

前缀和与差分

P10719

P10719 [GESP202406 五级] 黑白格

P10719

P10719 [GESP202406 五级] 黑白格

枚举所有的子矩形

P10719

[P10719 [GESP202406 五级] 黑白格]

枚举所有的子矩形

枚举矩形的左上角坐标 (a, c) 与右下角坐标 (b, d)。

也可以看成是枚举矩形的上下边界 a, b 和左右边界 c, d。

P10719

[P10719 [GESP202406 五级] 黑白格]

枚举所有的子矩形
枚举矩形的左上角坐标 (a,c) 与右下角坐标 (b,d)。

也可以看成是枚举矩形的上下边界 a,b 和左右边界 c,d。

共有 $O(n^2m^2)$ 个子矩形，使用二维前缀和统计黑色格子数。

P8649

[P8649 [蓝桥杯 2017 省 B] k 倍区间]

P8649

[P8649 [蓝桥杯 2017 省 B] k 倍区间]

连续子序列 $A_i, A_{i+1}, \cdots A_j (i \le j)$ 之和如何表示?

P8649

[P8649 [蓝桥杯 2017 省 B] k 倍区间]

连续子序列 $A_i, A_{i+1}, \cdots A_j (i \le j)$ 之和如何表示?

和为 K 的倍数如何表示?

P8649

[P8649 [蓝桥杯 2017 省 B] k 倍区间]

求有多少对 (i, j) 满足 $sum_j \bmod K = sum_{i-1} \bmod K$。

P8649

[P8649 [蓝桥杯 2017 省 B] k 倍区间]

求有多少对 (i,j) 满足 $sum_j \bmod K = sum_{i-1} \bmod K$。

枚举 j，统计有多少 $0 \le i < j$ 满足 $sum_i \bmod K = sum_j \bmod K$

使用桶即可。

P3909

P3909 异或之积

P3909

[P3909 异或之积]

暴力做法非常的明显，三重循环枚举 $1 \le i < j < k \le N$ 即可。

P3909

[P3909 异或之积]

暴力做法非常的明显，三重循环枚举 $1 \le i < j < k \le N$ 即可。

可以尝试进行第一次优化

两重循环枚举 $1 \le j < k \le N$，那么对应的式子变化为

$$\sum A_j \times A_k \times (\sum_{i=1}^{j-1} A_i)$$

设 $sum_i = \sum_{k=1}^{i} A_k$

$$\sum A_j \times A_k \times sum_{j-1}$$

P3909

P3909 异或之积

暴力做法非常的明显，三重循环枚举 $1 \le i < j < k \le N$ 即可。

再尝试进行第二次优化，如果只枚举 $1 \le k \le N$:

$$\sum_{k=1}^{N} A_k \times \left(\sum_{1 \le i < j < k} A_i \times A_j \right)$$

P3909

P3909 异或之积

暴力做法非常的明显，三重循环枚举 $1 \le i < j < k \le N$ 即可。

再尝试进行第二次优化，如果只枚举 $1 \le k \le N$：

$$\sum_{k=1}^{N} A_k \times \Big(\sum_{1 \le i < j < k} A_i \times A_j \Big)$$

考虑计算 $\sum_{1 \le i < j \le k-1} A_i \times A_j = f_{k-1}$

P3909

P3909 异或之积

暴力做法非常的明显，三重循环枚举 $1 \le i < j < k \le N$ 即可。

考虑计算 $\sum_{1 \le i < j \le k-1} A_i \times A_j = f_{k-1}$

$$f_k = f_{k-1} + A_k \times \sum_{i=1}^{k-1} A_i = f_{k-1} + A_k \times sum_{k-1}$$

P3909

P3909 异或之积

暴力做法非常的明显，三重循环枚举 $1 \le i < j < k \le N$ 即可。

考虑计算 $\sum_{1\le i<j\le k-1} A_i \times A_j = f_{k-1}$

$$f_k = f_{k-1} + A_k \times \sum_{i=1}^{k-1} A_i = f_{k-1} + A_k \times sum_{k-1}$$

如果只枚举 $1 \le k \le N$:

$$\sum_{k=1}^{N} A_k \times \left(\sum_{1\le i<j<k} A_i \times A_j \right) = \sum_{k=1}^{N} A_k \times f_{k-1}$$

P7404

[P7404 [JOI 2021 Final] とてもたのしい家庭菜園 4]

P7404

P7404 [JOI 2021 Final] とてもたのしい家庭菜園 4

考虑到题目要实现区间 $+1$，考虑利用差分序列。

P7404

[P7404 [JOI 2021 Final] とてもたのしい家庭菜園 4]

考虑到题目要实现区间 $+1$，考虑利用差分序列。

题目的要求相当于

$\forall 2 \le i \le k$ 满足 $B_i - B_{i-1} > 0$，并且 $\forall k+1 \le i \le N$ 满足 $B_i - B_{i-1} < 0$。

P7404

[P7404 [JOI 2021 Final] とてもたのしい家庭菜園 4]

考虑到题目要实现区间 +1，考虑利用差分序列。

题目的要求相当于

$\forall 2 \le i \le k$ 满足 $B_i - B_{i-1} > 0$，并且 $\forall k+1 \le i \le N$ 满足 $B_i - B_{i-1} < 0$。

设原本序列的差分序列为 $C_i = A_i - A_{i-1}$。

每次选择任意的一对 $1 \le i < j \le n+1$，让 $C_i + 1$ 并且 $C_j - 1$。

要让 C 的一个前缀为正数，剩下的后缀为负数。

P7404

[P7404 [JOI 2021 Final] とてもたのしい家庭菜園 4](#)

设原本序列的差分序列为 $C_i = A_i - A_{i-1}$。

每次选择任意的一对 $1 \le i < j \le n+1$，让 $C_i + 1$ 并且 $C_j - 1$。

要让 C 的一个前缀为正数，剩下的后缀为负数。

枚举 k，要将 $[2,k]$ 的 C 变为至少 1，后面的变为至多 -1。

$[2,k]$ 变得符合要求需要的 $+1$ 次数为 a，$[k+1,n]$ 变为符合要求的 -1 次数为 b，当前的这个 k 答案就是 $\max(a,b)$。

双指针/单调性

P8160

P8160 [JOI 2022 Final] 星际蛋糕 (Intercastellar)

P8160

[P8160 [JOI 2022 Final] 星际蛋糕 (Intercastellar)]

考虑 24 会被怎么划分。

P8160

[P8160 [JOI 2022 Final] 星际蛋糕 (Intercastellar)]

考虑 24 会被怎么划分。

$$\begin{array}{c} 24 \\ 12\ 12 \\ 6\ 6\ 6\ 6 \\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3 \end{array}$$

P8160

P8160 [JOI 2022 Final] 星际蛋糕 (Intercastellar)

考虑 24 会被怎么划分。

$$
\begin{array}{c}
24 \\
12\ 12 \\
6\ 6\ 6\ 6 \\
3\ 3\ 3\ 3\ 3\ 3\ 3\ 3
\end{array}
$$

由于划分出去的长度一定都一致，所以如果是偶数，也会一起被切分。

P8160

[P8160 [JOI 2022 Final] 星际蛋糕 (Intercastellar)]

考虑 24 会被怎么划分。

$$\begin{array}{c} 24 \\ 12\ 12 \\ 6\ 6\ 6\ 6 \\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3 \end{array}$$

由于划分出去的长度一定都一致，所以如果是偶数，也会一起被切分。

如果有 num 个长度为 len 的蛋糕，如果 len 为偶数，就会变成 $2num$ 个长度为 $\frac{len}{2}$ 的蛋糕。

P8160

[P8160 [JOI 2022 Final] 星际蛋糕 (Intercastellar)]

如果有 num 个长度为 len 的蛋糕，如果 len 为偶数，就会变成 $2num$ 个长度为 $\frac{len}{2}$ 的蛋糕。

可以把同一个蛋糕划分出来的小蛋糕打成一包，统计每一包的个数。

P8160

P8160 [JOI 2022 Final] 星际蛋糕 (Intercastellar)

如果有 num 个长度为 len 的蛋糕，如果 len 为偶数，就会变成 $2num$ 个长度为 $\frac{len}{2}$ 的蛋糕。

可以把同一个蛋糕划分出来的小蛋糕打成一包，统计每一包的个数。

题目所求即第 pos 包小蛋糕的长度，其中 pos 是最小的满足 $[1, pos]$ 这些包蛋糕个数加起来大于等于 X 的包数。

P8160

[P8160 [JOI 2022 Final] 星际蛋糕 (Intercastellar)]

如果有 num 个长度为 len 的蛋糕，如果 len 为偶数，就会变成 $2num$ 个长度为 $\frac{len}{2}$ 的蛋糕。

可以把同一个蛋糕划分出来的小蛋糕打成一包，统计每一包的个数。

题目所求即第 pos 包小蛋糕的长度，其中 pos 是最小的满足 $[1, pos]$ 这些包蛋糕个数加起来大于等于 X 的包数。

题目保证了 X 单增，因此可以使用一个单调移动的指针表示 pos

P10710

P10710 [NOISG2024 Prelim] School Photo

P10710

[P10710 [NOISG2024 Prelim] School Photo]

把全部的 $n \times s$ 个同学排好序，可以发现选择的一定是连续的一段。

（一个班的重复选了也没关系，只要这一段中包含了 n 个班级的同学）

（重复的不去就行了）

P10710

P10710 [NOISG2024 Prelim] School Photo

把全部的 $n \times s$ 个同学排好序，可以发现选择的一定是连续的一段。
（一个班的重复选了也没关系，只要这一段中包含了 n 个班级的同学）
（重复的不去就行了）

我们枚举最小值 i，对应的右端点/最大值随着最小值增大单调右移。
只要维持 $[i, pos]$ 内包含 n 个班级的同学并且 pos 是最小的。

哈希

P10468

[P10468 兔子与兔子]

P10468

[P10468 兔子与兔子]

求出 $[l_1, r_1]$ 和 $[l_2, r_2]$ 子串的哈希值判等。

P10468

P10468 兔子与兔子

哈希回顾：快速求出区间 $[l,r]$ 子串的哈希值

$$f_k = \sum_{i=1}^{k} b^{k-i} s_i = b^{k-1}a_1 + b^{k-2}a_2 + \cdots ba_{k-1} + a_k$$

$$f_k = b \times f_{k-1} + a_k$$

P10468

P10468 兔子与兔子

哈希回顾：快速求出区间 $[l,r]$ 子串的哈希值

$$[l,r] = b^{r-l}a_l + b^{r-(l+1)}a_{l+1} + \cdots + ba_{r-1} + a_r$$

P10468

P10468 兔子与兔子

哈希回顾：快速求出区间 $[l,r]$ 子串的哈希值

$$[l,r] = b^{r-l}a_l + b^{r-(l+1)}a_{l+1} + \cdots + ba_{r-1} + a_r$$

$$f_k = \sum_{i=1}^{k} b^{k-i}a_i = b^{k-1}a_1 + b^{k-2}a_2 + \cdots ba_{k-1} + a_k$$

$$f_k = b \times f_{k-1} + a_k$$

P10468

P10468 兔子与兔子

哈希回顾：快速求出区间 $[l,r]$ 子串的哈希值

$$f_k = \sum_{i=1}^{k} b^{k-i} a_i = b^{k-1}a_1 + b^{k-2}a_2 + \cdots b a_{k-1} + a_k$$

$$f_k = b \times f_{k-1} + a_k$$

$$[l,r] = b^{r-l}a_l + b^{r-(l+1)}a_{l+1} + \cdots + b a_{r-1} + a_r$$
$$= f_r - f_{l-1} \times b^{r-l+1}$$

P8630

[P8630 [蓝桥杯 2015 国 B] 密文搜索]

P8630

[P8630 [蓝桥杯 2015 国 B] 密文搜索]

考虑密码的所有排列的可能性，那么只要密码和某个长度为 8 的子串字母组成一致即可判等。

P8630

[P8630 [蓝桥杯 2015 国 B] 密文搜索]

考虑密码的所有排列的可能性，那么只要密码和某个长度为 8 的子串字母组成一致即可判等。

可以将每个长度为 8 的子串内部排序，再利用哈希统计出现次数

可以使用哈希表，也可以使用 map。

搜索

P7200

[P7200 [COCI2019-2020#1] Lutrija]

P7200

[P7200 [COCI2019-2020#1] Lutrija]

需要一些数学分析得到结论

P7200

P7200 [COCI2019-2020#1] Lutrija

需要一些数学分析得到结论

1. 肯定不会有重复元素

P7200

[P7200 [COCI2019-2020#1] Lutrija]

需要一些数学分析得到结论

1. 肯定不会有重复元素
2. 答案中相邻的两个数字要么有 2，要么差值为 2。

P7200

P7200 [COCI2019-2020#1] Lutrija

需要一些数学分析得到结论

1. 肯定不会有重复元素
2. 答案中相邻的两个数字要么有 2，要么差值为 2。

由结论 2 进一步推导，每次加入的数字只可能是 $A-2, A+2, 2, B-2, B+2$ 这几个数字之中的某一个。

P7200

[P7200 [COCI2019-2020#1] Lutrija]

需要一些数学分析得到结论

1. 肯定不会有重复元素
2. 答案中相邻的两个数字要么有 2，要么差值为 2。

由结论 2 进一步推导

每次加入的数字只可能是 $A-2$，$A+2$，2，$B-2$，$B+2$ 这几个数字之中的某一个。

P7200

P7200 [COCI2019-2020#1] Lutrija

需要一些数学分析得到结论

1. 肯定不会有重复元素
2. 答案中相邻的两个数字要么有 2，要么差值为 2。

由结论 2 进一步推导

每次加入的数字只可能是 $A-2$，$A+2$，2，$B-2$，$B+2$ 这几个数字之中的某一个。

使用 dfs 尝试填写即可。

B3624

B3624 猫粮规划

B3624

[B3624 猫粮规划]

进行搜索，按照每一份食物划分阶段，每份食物只有选/不选。

B3624

B3624 猫粮规划

进行搜索，按照每一份食物划分阶段，每份食物只有选/不选。

$O(2^n), n \le 40$ 过不了

B3624

[B3624 猫粮规划](#)

进行搜索，按照每一份食物划分阶段，每份食物只有选/不选。

$O(2^n), n \le 40$ 过不了

剪枝！

B3624

B3624 猫粮规划

进行搜索，按照每一份食物划分阶段，每份食物只有选/不选。

$O(2^n), n \leq 40$ 过不了

剪枝!

对于左右端点进行可行性剪枝:

左端点，如果把后续的食物全部算上也不够 l，那么一定无解。

右端点，如果后续什么都不选也已经超过了 r，那么一定无解。

分治

T580147

T580147 【第六周】关联扫描

T580147

T580147 【第六周】关联扫描

类似逆序对使用分治解决

T580147

[T580147 【第六周】关联扫描]

类似逆序对使用分治解决

solve(l,r) 表示计算区间 $[l,r]$ 内部的关联度之和。

一分为二，递归计算。

solve(l,mid)

solve(mid+1,r)

T580147

T580147 【第六周】关联扫描

`solve(l,r)` 表示计算区间 $[l,r]$ 内部的关联度之和。

一分为二，递归计算。

`solve(l,mid)`

`solve(mid+1,r)`

还需要计算的就是 $[mid+1,r]$ 能扫到的 $[l,mid]$ 范围内的雷达站次数之和。

T580147

[T580147 【第六周】关联扫描]

还需要计算的就是 $[mid+1,r]$ 范围内的雷达站能扫到的 $[l,mid]$ 范围内的雷达站次数之和。

按照 a_i 归并排序，计算 $[a_j-b_j,a_j+b_j]$ 范围内有多少个 a_i，二分查找即可。

其中 $i\in[l,mid],j\in[mid+1,r]$。

T580147

[T580147 【第六周】关联扫描]

还需要计算的就是 $[mid+1,r]$ 范围内的雷达站能扫到的 $[l,mid]$ 范围内的雷达站次数之和。

按照 a_i 归并排序，计算 $[a_j-b_j,a_j+b_j]$ 范围内有多少个 a_i，二分查找即可。

其中 $i\in[l,mid],j\in[mid+1,r]$。

分治边界即为 $l=r$，此时直接返回。

T580148

T580148 【第六周】刷围栏

T580148

T580148 【第六周】刷围栏

观察结论:

1. 横着涂的话一定是尽可能长的涂色长度。

T580148

[T580148 【第六周】刷围栏]

观察结论:

1. 横着涂的话一定是尽可能长的涂色长度。
2. 如果在一段栅栏里横着涂，一定是把最矮的涂满了。

T580148

[T580148 【第六周】刷围栏]

1. 横着涂的话一定是尽可能长的涂色长度。
2. 如果在一段栅栏里进行过横着涂，一定是把最矮的涂满了。

设计分治算法

`solve(l,r)` 表示计算区间 $[l,r]$ 栅栏的最小涂色次数。

如果不横着涂，那么答案就是 $r-l+1$。

T580148

T580148 【第六周】刷围栏

1. 横着涂的话一定是尽可能长的涂色长度。
2. 如果在一段栅栏里进行过横着涂，一定是把最矮的涂满了。

设计分治算法

`solve(l,r)` 表示计算区间 $[l,r]$ 栅栏的最小涂色次数。

如果不横着涂，那么答案就是 $r-l+1$。

如果横着涂，那么就把区间内栅栏的高度都减掉最小值。以得到的 0 作为划分依据把区间 $[l,r]$ 划分成几个子区间递归分治。

T580148

T580148 【第六周】刷围栏

1. 横着涂的话一定是尽可能长的涂色长度。
2. 如果在一段栅栏里进行过横着涂，一定是把最矮的涂满了。

设计分治算法

`solve(l,r)` 表示计算区间 $[l,r]$ 栅栏的最小涂色次数。

如果不横着涂，那么答案就是 $r-l+1$。

如果横着涂，那么就把区间内栅栏的高度都减掉最小值。以得到的 0 作为划分依据把区间 $[l,r]$ 划分成几个子区间递归分治。

边界是 $l=r$，此时答案为 1。

位运算

T580152

[T580152 【第六周】或或和之和]

T580152

T580152 【第六周】或或和之和

拆位考虑，假设当前在考虑 2^k 的这一位。

那么就是要计算有多少个区间或的结果在这一位为 1。

T580152

T580152 【第六周】或或和之和

拆位考虑，假设当前在考虑 2^k 的这一位。
那么就是要计算有多少个区间或的结果在这一位为 1。

枚举右端点 r，考虑有效左端点范围。

T580152

T580152 【第六周】或或和之和

拆位考虑，假设当前在考虑 2^k 的这一位。
那么就是要计算有多少个区间或的结果在这一位为 1。

枚举右端点 r，考虑有效左端点范围。

$[1,p]$ 其中 p 是最大的满足 $p \le r$ 并且 a_p 在 2^k 这一位为 1。

倍增

T580149

T580149 【第六周】阻拦乘积

T580149

[T580149 【第六周】阻拦乘积]

先考虑这个问题

给定一个位置 x，找到这次开炮会被哪座山拦住，如果能飞到 n 那么就是被第 $n+1$ 座山拦住。

T580149

T580149 【第六周】阻拦乘积

先考虑这个问题

给定一个位置 x，找到这次开炮会被哪座山拦住，如果能飞到 n 那么就是被第 $n+1$ 座山拦住。

https://www.luogu.com.cn/problem/P2866

与第四周例题一样。

T580149

T580149 【第六周】阻拦乘积

对于每个询问 x，找到被拦住的山 y，那么只要求出 x 往后连续 $y-x$ 座山高度的乘积 $\bmod 10000$ 即可。

T580149

[T580149 【第六周】阻拦乘积]

对于每个询问 x，找到被拦住的山 y，那么只要求出 x 往后连续 $y-x$ 座山高度的乘积 $\bmod 10000$ 即可。

以 $f[k][x]$ 表示从第 x 座山开始连续 2^k 座山高度的乘积 $\bmod 10000$ 的结果（$[x,x+2^k-1]$）。

T580149

T580149 【第六周】阻拦乘积

对于每个询问 x，找到被拦住的山 y，那么只要求出 x 往后连续 $y-x$ 座山高度的乘积 $\bmod 10000$ 即可。

以 $f[k][x]$ 表示从第 x 座山开始连续 2^k 座山高度的乘积 $\bmod 10000$ 的结果（$[x, x+2^k-1]$）。

对 $y-x$ 进行二进制分解即可。

T580151

T580151 【第六周】树上阻拦乘积

T580151

T580151 【第六周】树上阻拦乘积

同前一题，我们先考虑求，从 x 向上跳多少步会被拦住。

T580151

T580151 【第六周】树上阻拦乘积

同前一题，我们先考虑求，从 x 向上跳多少步会被拦住。

以 $mx[k][x]$ 表示从 x 往上 2^k 个节点范围内高度的最大值。

```
int pos=fa[0][x];
for(int k=J;k>=0;k--)
{
    if(dep[pos]>=(1<<k)&&mx[k][pos]<a[x])
        pos=fa[k][pos];
}
```

T580151

T580151 【第六周】树上阻拦乘积

同前一题，我们先考虑求，从 x 向上跳多少步会被拦住。

以 $mx[k][x]$ 表示从 x 往上 2^k 个节点范围内高度的最大值。

```
int pos=fa[0][x];
for(int k=J;k>=0;k--)
{
    if(dep[pos]>=(1<<k)&&mx[k][pos]<a[x])
        pos=fa[k][pos];
}
```

以此可以求出会被点pos拦住。

T580151

T580151 【第六周】树上阻拦乘积

同前一题，我们先考虑求，从 x 向上跳多少步会被拦住。

以 $mx[k][x]$ 表示从 x 往上 2^k 个节点范围内高度的最大值。

```
int pos=fa[0][x];
for(int k=J;k>=0;k--)
{
    if(dep[pos]>=(1<<k)&&mx[k][pos]<a[x])
        pos=fa[k][pos];
}
```

以此可以求出会被点pos拦住。

那么从x向上跳dep[x]-dep[pos]步所有山高度的乘积即为答案。

T580151

T580151 【第六周】树上阻拦乘积

求出会被点 pos 拦住。

那么从 x 向上跳 $dep[x]-dep[pos]$ 步所有山高度的乘积 $\bmod 10000$ 即为答案。

以 $mul[k][x]$ 表示从点 x 开始往上 2^k 级节点山高度乘积 $\bmod 10000$

T580151

T580151 【第六周】树上阻拦乘积

求出会被点 pos 拦住。

那么从 x 向上跳 $dep[x]-dep[pos]$ 步所有山高度的乘积 $\bmod 10000$ 即为答案。

以 $mul[k][x]$ 表示从点 x 开始往上 2^k 级节点山高度乘积 $\bmod 10000$

对 $d=dep[x]-dep[pos]$ 进行二进制分解即可。