



第一周 作业讲评

基础-提高衔接计划

览遍千秋

2025-07-19



www.luogu.com.cn

课前提示

- 上课的时候专心听讲解，**不要跟着老师抄代码**，下课后独立完成。
- 不使用 AI 做题，AI 会做不等于自己会。
- 不抄袭题解（含对照题解抄一遍），抄对不等于会做。
- 看完题解后，关闭题解独立练习。
- 练习中途遇到问题，应当分析题目及自己的思路，而非回忆题解或再次参考题解。
- 做过的题在课后需要重新独立完成，不参考老师的课件、代码，不参考自己以前的代码。

分数统计

- A: <https://www.luogu.com.cn/problem/T631941>
- B: <https://www.luogu.com.cn/problem/T631945>

分数统计

- 要求有 x 名不少于 x 分的学生
- 答案最大为 n
- 从大到小枚举 x , 遍历 $a[i](1 \leq i \leq n)$
- 统计计算 $a[i] \geq x$ 的数目 c
- 找到的第一个 x 满足 $c \geq x$ 即为答案
- 时间复杂度 $O(n^2)$ 或 $O(n \log n)$

分数统计

5

5 5 5 5 3

- 这份代码输出 3，正确结果 4
- 答案不必是某名同学的分数
- 只是 5 分的人数不够
- 降低人数要求就能满足

```
int main() {  
    scanf("%d",&n);  
    for(int i=1;i<=n;i++)  
        scanf("%d",&a[i]);  
    sort(a+1,a+n+1,cmp);  
    for(int i=1;i<=n;i++) {  
        if(i>=a[i]) {  
            mx=max(mx,a[i]);  
        }  
    }  
    printf("%d",mx);  
    return 0;  
}
```

数的一万种排序方法

- A: <https://www.luogu.com.cn/problem/T631942>
- B: <https://www.luogu.com.cn/problem/T631946>

数的一万种排序方法

- 预处理每个数的封闭图形个数
- 结构体 sort
- 或者直接在 sort 的 cmp 函数中算 封闭图形个数

数的一万种排序方法

- Write clean code.
- 用 $arr[x]$ 表示数码 x 中封闭图形的数目
- 而不是很长的 if-else

游戏签到

- A: <https://www.luogu.com.cn/problem/T631943>
- B: <https://www.luogu.com.cn/problem/T631947>

游戏签到

- 弄清题设结算顺序:
- 如果当天没有签到, 简单的累计连续未签到天数
- 如果当天签到了, 先结算未签到的扣除, 并清空连续未签到天数, 再增加连续签到天数
- 注意 corner case: 连续未签到天数为 0, 不能计算 2^{-1}

拓展三子棋

- A: <https://www.luogu.com.cn/problem/T631944>
- B: <https://www.luogu.com.cn/problem/T631948>

拓展三子棋

- 棋盘出现连续 3 个棋子获胜
- 如何枚举连续 3 个棋子?
- 方法 1:
- 枚举一端
- 方法 2:
- 枚举中心棋子

拓展三子棋

- 如何表示枚举的方向？
- 增量数组
- 而不是 4/8 个 if

B4020 两座城市的 543 千米

- <https://www.luogu.com.cn/problem/B4020>

B4020 两座城市的 543 千米

- M 次高铁，每次高铁相互独立
- 给出每一次高铁经过的站点，问是否先经过 a 再经过 b
- 用变量 Va 表示 a 是否出现，以及出现的位置
- 用变量 Vb 表示 b 是否出现，以及出现的位置
- 如果 Va, Vb 都出现了，并且 $Va < Vb$ ，就说明 a 在 b 前面

B4146 本俗妙手不如举手

- <https://www.luogu.com.cn/problem/B4146>

B4146 本俗妙手不如举手

- 枚举 l ，修改 l 开始的 k 场比赛结果
- 用下标表示就是 $[l, l + k - 1]$
- 循环计算每个人的得分
- 回溯修改

P8897 Cow College

- <https://www.luogu.com.cn/problem/P8897>

P8897 Cow College

- 结论：最终确定的学费一定是 c_i 中的某一个值

P8897 Cow College

- 按 c 从大到小排序
- 如果价格确定为 c_i , 那么有 i 名奶牛愿意支付
- 收益为 $i \times c_i$

类型与运算顺序

- i 与 $c[i]$ 均为 `int` 型

```
int k = i * c[i];
```

```
long long k = i * c[i];
```

```
long long k = 1ll * i * c[i];
```

P9010 Leaders

- <https://www.luogu.com.cn/problem/P9010>

P9010 Leaders

- 奶牛 i 需要满足以下两个条件之一才能成为领导：
- ① $i \sim E[i]$ 包含了全部同品种牛
- ② $i \sim E[i]$ 包含了另一品种的领导
- 满足条件 ① 的牛，每个品种最多只有一头
- 两种牛的领导必有一前一后，前者可以是 ①②，后者只能是 ①
- 找到可能是 ① 的牛
- 再找到符合 ② 的另一种牛

P9010 Leaders

- 找到可能是 ① 的牛
- 如果直接循环计算，时间复杂度为 $O(n^2)$
- 不可接受

- 找到本品种最后一头牛，位置记为 r
- 找到本品种的第一头牛，位置记为 l
- 将 $E[l]$ 与 r 比较
- 如果 $E[l] \geq r$ ， l 就是符合 ① 的牛
- 否则该品种不存在 ① 领导

P9010 Leaders

- 领导对只有 ①① 与 ②① 两种可能
- 接下来，枚举每一头牛 i ，如果它是本品种的 ①，则直接答案增加，否则判断其可不可以作为 ②
- 要满足：
 - 另一品种的 ① 存在
 - 另一品种的 ① 在 i 后面
 - 另一品种的 ① 在 $E[i]$ 前面
- 最终输出的时候答案需要 -1，去重 ①①

B4138 洗牌

- <https://www.luogu.com.cn/problem/B4138>

B4138 洗牌

- 原数组为 a ，洗牌后的数组为 b
- 如果 i 是奇数， $b[i] = a\left[\frac{i+1}{2}\right]$
- 如果 i 是偶数， $b[i] = a\left[\frac{i+n}{2}\right]$
- 每次操作完毕后，将 b 重新拷贝给 a

B3708 神树大人挥动魔杖

- <https://www.luogu.com.cn/problem/B3708>

B3708 神树大人挥动魔杖

- 最小的 n 位数是 10^{n-1}
- 最大的 n 位数是 $10^n - 1$
- 在 $10^{n-1} \sim 10^n - 1$ 的范围内枚举 i
- 按 $i \bmod k$ 的值划分进桶
- 时间复杂度 $O(10^n)$

B3708 神树大人挥动魔杖

- 另外有一种 $O(k)$ 的做法
- $i \bmod k$ 的余数是循环的
- $\dots, 0, 1, 2, \dots, k-2, k-1, 0, \dots$
- 完整循环数目为 $\left\lfloor \frac{len}{k} \right\rfloor$
- 其中 len 为区间长度, $len = 10^n - 10^{n-1} = 9 \times 10^{n-1}$
- 再补齐头尾即可
- 时间复杂度 $O(k)$

B4024 保持连接的方式

- <https://www.luogu.com.cn/problem/B4024>

B4024 保持连接的方式

- 使用三维数组 $a[][][]$ 来模拟小木箱。 $a[x][y][]$ 记录 (x, y) 存放的日记编号。用二维数组 $top[][]$ 来记录每一个小格子已经存放的日记数目， $top[x][y]$ 表示 (x, y) 已经存放的日记数目。
- 对于输入的一篇日记，需要存放到 (x, y) ，编号为 id ，首先检查 $top[x][y]$ 的值

B4024 保持连接的方式

- 若 $top[x][y] = k$
- 则说明 (x, y) 已经放满了，我们需要遍历 $a[x][y][]$ ，记录下最小值与最小值的位置，假设最小值为 $a[x][y][pos]$ ， pos 即为位置，那么需要移动的日记份数即为 $k - pos$ 。
- 之后，进行数组的移动，将 $i > pos$ 的 $a[x][y][i]$ 移动到 $a[x][y][i - 1]$ 。最后将新的日记存入 (x, y) ，即 $a[x][y][k]$ 赋值为 id 。

B4024 保持连接的方式

- 若 $top[x][y] \neq k$
- 直接将 $top[x][y]$ 增加 1, 将 $a[x][y][top[x][y]]$ 赋值为 id 。

B3818 电脑中了满屏粉兔病毒

- <https://www.luogu.com.cn/problem/B3818>

B3818 电脑中了满屏粉兔病毒

- 样例是正确的

洛谷

B3818 电脑中了满屏粉兔病毒

- 枚举每一只粉兔
- 向四个方向枚举，看是否存在异性粉兔
- 如何向四个方向枚举？
- 使用增量数组与增量长度
- 时间复杂度 $O(r^2c + rc^2)$

B3818 电脑中了满屏粉兔病毒

- 这个时间复杂度非常紧张了
- 主要的瓶颈在于向四个方向枚举
- 预处理
- 用 $f[0/1][i][j][k]$ 表示 (i, j) 在增量数组方向 k 的格子上是否存在性别为 0/1 的粉兔
- 可以免去向四个方向枚举
- 时间复杂度 $O(rc)$

B4040 黑白方块

- <https://www.luogu.com.cn/problem/B4040>

B4040 黑白方块

- 枚举 4×4 方格的左上角
- 将目标图形存储在一个二维数组中
- 循环检查这个 4×4 方格是否与目标相符

B3675 军训

- <https://www.luogu.com.cn/problem/B3675>

B3675 军训

- 这个公式到底是什么意思

$$\frac{1}{m} \times \sum_{j=1}^m \left(a_{i,j} - \frac{\sum_{k=1}^m a_{i,k}}{m} \right)^2 + \frac{1}{m} \times \sum_{j=1}^m \left(b_{i,j} - \frac{\sum_{k=1}^m b_{i,k}}{m} \right)^2$$

B3675

- 如何得到一个合法的方案
- 其实就是输出一个排序中，每一次交换
- 题目给了上限 n^2 次
- 冒泡排序
- 选择排序
- 注意输入格式

P8899 Reverse Engineering

- <https://www.luogu.com.cn/problem/P8899>

P8899 Reverse Engineering

- 核心问题：一个 if 语句会产生什么效果？
- 按某个变量将数据撕裂为两部分

b[0]	b[1]	b[2]	target
1	0	1	1
1	1	1	0
0	1	1	1
0	1	1	0
1	0	1	1

P8899 Reverse Engineering

- `if(b[0] == 1) {`

b[0]	b[1]	b[2]	target	划分入 if 的 target 必须 一致
1	0	1	1	
1	1	1	0	
1	0	1	1	

- `}`
- `else {`

b[0]	b[1]	b[2]	target	划分入 else 的数据可以 再分割
0	1	1	1	
0	1	1	0	

- `}`

P8899 Reverse Engineering

- 如果我们现在按照 $b[0]$ 划分.....

b[0]	b[1]	b[2]	target
1	0	1	1
1	1	1	0
1	0	1	1
0	1	1	1
0	1	1	0

- 如果尝试了每一个变量，都划分失败
- 有人在说谎

P8899 Reverse Engineering

- 如果我们现在按照 $b[1]$ 划分.....

b[0]	b[1]	b[2]	target
1	0	1	1
1	0	1	1
1	1	1	0
0	1	1	1
0	1	1	0

- 下一条语句可以是
- ... else if($b[1] == 0$) return 1;
- else

P8899 Reverse Engineering

- 最后的一步：
- 按照某一种变量的某一个值划分
- 可以全部纳入一个语句块
- 作为最终的 else-return 语句

P8899 Reverse Engineering

- Conclusion & Steps:
- 枚举 $b[i](0 \leq i < n)$ ，尝试用 $b[i]$ 的值将剩余数据划分为两部分（称为 0 部分与 1 部分）。
- 检查两部分数据，是否存在至少一部分中的数据 target 都相同，若不存在，枚举下一个。如果枚举完仍没有找到，则报告无解。
- 选择相同的一部分，作为这一条语句。从 $b[0]$ 重新开始枚举。直至只能划分为一个部分，且 target 相同。则有解，并结束程序。

Q: What is a good code style for OI?

- 易于阅读和调试
- 函数化与模块化
- 思路化
- 适当的封装