

元旦激光炮 解题报告

湖北省武汉市第二中学 吕凯风

1 试题来源

原创。可以在这里找到：<http://uoj.ac/problem/52>。

2 试题大意

刚刚越过绝境长城，只见天空中出现了炫目的光芒——圣诞老人出现了。

元旦三侠立刻进入战斗。生蛋侠、圆蛋侠和零蛋侠分别有 n_a, n_b, n_c 个激光炮。生蛋侠的激光炮的威力分别为 $a_0, a_1, \dots, a_{n_a-1}$ ，圆蛋侠的激光炮的威力分别为 $b_0, b_1, \dots, b_{n_b-1}$ ，零蛋侠的激光炮的威力分别为 $c_0, c_1, \dots, c_{n_c-1}$ 。

元旦三侠的激光炮的威力已经按从小到大的顺序排好了，即 $a_{i-1} \leq a_i$ ， $b_{i-1} \leq b_i$ ， $c_{i-1} \leq c_i$ 。

由于元旦三侠精力有限，他们得废弃掉 k 个激光炮。为了更好地进行战斗，他们决定废弃掉威力前 k 小的激光炮。

赶快帮助元旦三侠，让激光炮投入战斗吧！你只需要告诉他们威力第 k 小的激光炮威力是多少。

你需要编写一个函数 `query_kth`，以确定威力值第 k 小的激光炮威力值是多少。

- `query_kth(n_a, n_b, n_c, k)`
 - `n_a`: 生蛋侠拥有的激光炮数目 n_a 。保证 $n_a \geq 0$ 。
 - `n_b`: 圆蛋侠拥有的激光炮数目 n_b 。保证 $n_b \geq 0$ 。
 - `n_c`: 零蛋侠拥有的激光炮数目 n_c 。保证 $n_c \geq 0$ 。
 - `k`: 要查询的排名 k 。保证 $1 \leq k \leq n_a + n_b + n_c$ 。

你可以调用三个函数 `get_a`、`get_b`、`get_c` 以帮助你确定第 k 小的激光炮。我们会根据你调用这三个函数的总次数评分。

- `get_a(i)` 将返回 a_i 。如果 i 在 $0 \leq i < n_a$ 之外，该函数将返回 2147483647。
- `get_b(i)` 将返回 b_i 。如果 i 在 $0 \leq i < n_b$ 之外，该函数将返回 2147483647。
- `get_c(i)` 将返回 c_i 。如果 i 在 $0 \leq i < n_c$ 之外，该函数将返回 2147483647。

在一组测试数据中，`query_kth` 只会被调用一次。

共 10 个测试点，每个测试点 10 分。设你的程序 `get_a`、`get_b`、`get_c` 函数的调用次数为 t 。当 $t \leq 100$ 时得 10 分，否则当 $t \leq 2000$ 时得 6 分，否则不得分。

编号	特殊限制
1	$n_a, n_b, n_c \leq 30$
2	$n_c = 0$
3	
4	
5	无
6	
7	
8	
9	
10	

对于所有测试点， $0 \leq n_a, n_b, n_c \leq 10^5$ ， $1 \leq a_i, b_i, c_i \leq 10^9$ 。

时间限制：1s

空间限制：256MB

3 算法介绍

4 算法一

我们可以用 $n_a + n_b + n_c$ 次询问把整个数组给询问出来，然后排个序求出第 k 小。

可以通过 1 号测试点获得 10 分。

5 算法二

显然可以二分答案嘛！二分第 k 小的值，然后在三个数组里二分统计小于等于这个值的有多少个。

在长度为 n 的数组里二分需要调用 $\lceil \log_2 n \rceil$ 次，而答案的范围是 10^9 需要二分 30 次，所以总次数最坏是 $30 \times 17 \times 3 = 1530$ 次。显然这离 2000 次调用的限制差得远。

可以拿到每个测试点的 6 分获得 60 分。

如果你特判掉了 1 号测试点使用暴力，那么可以拿到 1 号测试点的 10 分获得 64 分。

6 算法三

注意到有三个超良心的测试点，其中一个数组是空的，只有两个数组。我们可以假想这两个数组的长度为 k ，然后这里有一个非常有趣的给两个数组取中位数的算法可以解决此问题。

现在手上有两个数组 a, b 长度均为 n 且 n 为偶数。我们记 $i = \lfloor n/2 \rfloor$, $j = \lceil n/2 \rceil$ 。（这里假设数组下标从 1 开始）

我们比较一下 a_i 和 b_j 的大小。如果 $a_i < b_j$ ，那么：

- a_1 到 a_i 均比 b_j 到 b_n 要小，这说明 a 中有 $n - i$ 个元素， b 中有 $n - j + 1$ 个元素比他们大，总共 $n + 1$ 个元素比他们大。
- 另一方面，这说明 a 中有 i 个元素， b 中有 j 个元素比 b_{j+1} 到 b_n 要小，总共 n 个元素比他们小。

所以，我们可以去掉 a 的前 i 个元素和 b 的后 i 个元素而中位数不变，然后递归下去就行了。对于 $a_i \geq b_j$ 的情况可以类比。最后递归到 $n = 1$ 时，暴力下就行了。

这样只用 $2 \times (\lceil \log_2 n \rceil + 1)$ 次调用即可解决问题。对于 $k \leq 2 \times 10^5$ ，调用次数只有 $2 \times 18 = 36$ 次。

结合算法二可以获得 72 分。如果再结合算法一可以获得 76 分。

7 算法四

当我知道有个算法三的时候我惊呆了……其实我想表演的是另一个神奇的算法。

首先我们把数组的长度扔进垃圾桶！我们想象数组是无穷长的，即在原数组后面补无穷个无穷大。

然后我们取 $l = \lfloor k/3 \rfloor$ 。我们来比较 a_l, b_l, c_l 。我们假设 a_l 是这三个数中最小的。

那么我们看看最多有多少个数比 a_l 小，为了说话方便我们无视两个数相等的情况。 a 数组中肯定是 $l-1$ 个， b 数组中由于 b_l 比 a_l 大，所以最坏情况是 b_1 到 b_{l-1} 都比 a_l 小， c 数组也是一样。于是最多 $3l-3$ 个比 a_l 小。所以 a_l 的排名至多是 $3l-2 < k$ 。

于是我们可以甩掉 a 数组的前 l 个元素然后 $k = k - l$ ，递归下去。注意到每次 k 的大小都减少了三分之一，到最后 $k \leq 2$ 时我们可以用暴力 6 次调用解决问题。总调用次数满足递推式 $T(n) = T(n - \lfloor n/3 \rfloor) + 3$ ，所以会调用次数不超过 $\lceil \log_{3/2} k \rceil + 4$ 。

对于 $k \leq 3 \times 10^5$ ，调用次数只有 96，足以解决此题。

可以通过所有测试点获得 100 分。

8 数据生成方式

直接随机生成即可。

9 参考程序

- `kth.cpp` 是参考程序。
- `kth_orz.cpp` 是算法二。