

# CST2020 3-3 kth

## 题目描述

有三个数组  $a[1..n]$ 、 $b[1..n]$  和  $c[1..n]$ ，你需要在所有数对  $(x, y, z)$  ( $1 \leq x, y, z \leq n$ ) 中，找到  $a[x] + b[y] + c[z]$  的值第  $k$  小的那对，如果有多对，找出任何一对均可。

在本题中，你不能直接访问这三个数组中的元素，而只能给出两种方案（两个数对），通过我们提供的函数接口，比较它们的值的大小。

## 接口说明

我们提供了一个交互库，包括一个头文件 `kth.h` 和一个库文件 `kth_lib.cpp`。头文件包含了 `compare` 函数和 `get_kth` 函数的定义，库文件包含了 `compare` 函数的实现。

编译时，我们会将库文件和你的程序一起编译和链接，因此你只需在你的程序中包含头文件，并实现 `get_kth` 函数。

以下是各函数的声明：

```
void get_kth(int n, int k, int *x, int *y, int *z);
```

这是 `get_kth` 函数的声明。你需要在你的代码中实现该函数。其中  $n$  和  $k$  的含义见题目描述。当你找到第  $k$  小的数对后，需要将该数对的三个值分别存在  $*x$ 、 $*y$  和  $*z$  中，即指针  $x$ 、 $y$  和  $z$  所指向的地址中。

```
int compare(int x, int y, int z, int u, int v, int w);
```

这是 `compare` 函数的声明。我们的交互库中已经实现了该函数，你可以在你的程序中调用它。传入的  $x$ 、 $y$ 、 $z$  和  $u$ 、 $v$ 、 $w$  表示两个数对  $(x, y, z)$  和  $(u, v, w)$ ，当  $a[x] + b[y] + c[z] < a[u] + b[v] + c[w]$  时，返回 1，否则返回 0。

如果你传入的  $x$ 、 $y$ 、 $z$  和  $u$ 、 $v$ 、 $w$ ，不在  $1 \sim n$  的范围内，则你的程序将会被判为 “Wrong Answer”。

`compare` 函数的实现是  $O(1)$  的，并且你可以认为每秒钟至少能调用  $5e7$  次该函数。

## 测试说明

我们的交互库的具体实现如下：

首先从标准输入中读入两个整数  $n, k$ ，具体含义见“题目描述”。

然后以某种方式读入并生成  $a$ 、 $b$ 、 $c$  数组中的数，这一步对于我们提供的交互库和 OJ 上评测时用的交互库可能不同，但这不会影响你做题。

接着调用 `get_kth` 函数。

在你实现的 `get_kth` 函数中，可以调用 `compare` 函数来对两个数对进行比较，如果传入的参数不在  $1 \sim n$  的范围内，你的程序会被判为 “Wrong Answer”。

在 `get_kth` 函数返回后，我们会输出你的方案  $(x, y, z)$  的值，即  $a[x] + b[y] + c[z]$ 。

在 OJ 上评测时采用全文比较，所以只有当你的程序正确结束，并答案正确时，才能被判为正确。

如果你需要在本地测试你的程序，你可以使用 `g++ kth_lib.cpp your_code.cpp -O2 -o kth.exe` 编译你的程序，再运行 `kth.exe` 来测试。

你需要严格按照输入格式向交互库输入数据，否则不保证交互库正常运行。

我们还提供了一个样例程序 `kth_sample.cpp`，它会在  $n * n * n$  种方案中根据某种规则挑一个，不保证答案正确。你可以参考这个程序来理解交互库和解题，或者不参考，这将与你的得分无关。

请注意，你只需提交实现了 `get_kth` 函数的源程序文件，不要提交其他任何文件，否则可能会导致编译错误。

下载交互库和样例程序 (attachment/8ca9/8ca9ae8a1754b32da62682c58c24e6144091473b.zip)

## 输入

对于我们提供的交互库，输入格式如下：

第一行两个整数  $n, k$ ，具体含义见“题目描述”。

接下来一行  $n$  个整数，第  $i$  个整数表示  $a[i]$ 。

接下来一行  $n$  个整数，第  $i$  个整数表示  $b[i]$ 。

接下来一行  $n$  个整数，第  $i$  个整数表示  $c[i]$ 。

输入每个整数都不小于 1，且不大于  $1e8$ 。

## 输出

对于我们提供的交互库，当你的程序正确结束时，会输出你的方案  $(x, y, z)$  的值，即  $a[x] + b[y] + c[z]$ ，否则输出一些错误信息，具体见交互库源程序 `kth_lib.cpp`。

## 输入样例

```
3 6
3 1 5
4 2 1
200 300 100
```

请不要尝试在你的程序中读取标准输入或其他文件。

## 输出样例

```
106
```

这是该输入样例的正确输出。最小的几个方案的值分别为 102, 103, 104, 105, 105, 106, 107 ...

请不要尝试在你的程序中向标准输出或其他文件中输出信息。

## 数据范围

共有 20 个测试点。对于任何测试点， $1 \leq k \leq n * n * n$ 。每个测试点的数据范围如下：

测试点编号	$n$ 不超过	$k$ 不超过
1	5	10

2	10	20
3	100	300
4	1000	3000
5	10000	30000
6	20000	60000
7	40000	120000
8	70000	210000
9	100000	300000
10	500000	2000000
11	100	10000
12	1000	1000000
13	10000	2000000
14	30000	2000000
15	100000	2000000
16	200000	600000
17	300000	1000000
18	500000	2000000
19	300000	2000000
20	500000	2000000

## 资源限制

时间限制：3 sec

空间限制：256 MB

## 提示

需要学到优先级队列再解此题。

[ n 特别大，枚举所有方案好像不太可能 ]

[ 如果只有一个数组，要求第 k 小，可以怎么求？ ]

[ 如果有两个数组，该怎么求？ ]

[ 如果有三个数组呢？ ]

[ 堆 ]