

求升序矩阵中第k大的 数字

陈星

题目描述

给定一个 $N \times M$ 的矩阵，从左到右，从上到下都为升序排序，求矩阵中第 k 大的数字。

Example:

```
matrix = [  
    [ 1, 5, 11],  
    [10, 11, 12],  
    [11, 13, 15]  
],  
k = 8,
```

```
return 13.
```

题目来源: Leetcode 378

解法1

思路：将矩阵放入一维数组中，然后对数组排序，找出第k大元素

时间复杂度： $O(N*M*\log(N*M))$

空间复杂度： $O(N*M)$

解法2

思路：解法1存在浪费，只需要找到第k大元素，并不需要排序，找第k大元素有2种常规方式：快排思想和**堆排**思想。

堆排也有2种常用思维：

- 全部放入堆，堆从小到大排序，结束后然后pop k次。

时间复杂度：

$O(N*M*\log(N*M))$ 。pop复杂度是 $O(k*\log(N*M))$ ，但是放入时间复杂度是 $O(N*M*\log(N*M))$ ，所以取大值。

空间复杂度：

$O(N*M)$

- 挨个放入堆，堆从大到小排序，堆超过k个则pop。最后堆中剩k个，全部pop掉，取最后一个。

时间复杂度：

$O(N*M*\log(k))$

空间复杂度：

$O(K)$

解法3

- **分析：** 以上两种解法没有利用好原来矩阵的从左到右，从上到下有序的特性，随便一个无序矩阵都一样。
- **思路：** 根据的堆的想法，先将矩阵每行行首元素压入堆，堆从小到大排序，挨个pop，每次pop后压入对应元素右边那一列的元素。
- **举例：**
- **时间复杂度：** $O(\max\{k,n\} \cdot \log(m))$ 。其中 $O(k \cdot \log(m))$ 是pop的时间复杂度， $O(n \cdot \log(m))$ 是压入的时间复杂度。
- **空间复杂度：** $O(\log(m))$

解法4

给定target，如何找出矩阵中小于target的元素
的个数？

方法：Z字形遍历

解法：从右上角($i=0, j=M$)开始，如果 $a[i][j]<target$ ，则 $i+1$ ；如果 $a[i][j]>target$ ，则 $j-1$ ；相等则视题目约束而定。行号变动时， $sum+j$ 。

举例：target=7

1	5	6	7
2	6	8	9
6	12	15	20

初始：($i=0, j=3$), $sum=0$
($i=1, j=3$), $sum+4$
($i=1, j=2$)
($i=1, j=1$)
($i=2, j=1$), $sum+2$
($i=2, j=0$),
($i=3>N$), $sum+1$
结束： $sum=7$

- 通常，在寻找小于target的时候，记录sum_lt和sum_le，前者表示小于target的个数，后者表示小于等于target的个数。如果sum_lt>k，则移动二分右边界，如果sum_le<k，则移动二分左边界；剩下的情况，表示找到返回。

伪代码

```
le = min, ri = max;
while (le < ri) {
    mid = le + (ri - le) / 2;
    sum_le, sum_lt = judge(mid);
    if sum_lt > K {
        ri = mid;
    }
    else if sum_le < K {
        le = mid;
    }
    else {
        // mid is the result
    }
}
```

解法4

- **分析：**有了判定小于target个数的方法后，target怎么定？
- **思路：**通常情况，二分法。二分区间： $[\min=a[0][0], \max=a[N-1][M-1]]$
- **时间复杂度：** $O((N+M)*\log(\max-\min))$ 。其中 $O(N+M)$ 是判定小于target个数的时间复杂度， $\log(\max-\min)$ 是二分。
- **空间复杂度：** $O(1)$

解法5

- 方法: <http://www.cse.yorku.ca/~andy/pubs/X+Y.pdf>
- 时间复杂度: $O(n)$
- 空间复杂度: $O(n)$

举一反三

- 给定的题目可能会变形，输入不是矩阵，而是别的形式，比如两个数组的拼接形式。
- Leetcode 373
- 题目描述：
 - 给定2个升序数组a, b，从a,b中各挑出一个元素，返回加和最小的k个元素。
- 举例：

Given a = [1,7,11], b = [2,4,6], k = 3
Return: [1,2],[1,4],[1,6]
The first 3 pairs are returned from the sequence:
[1,2],[1,4],[1,6],[7,2],[7,4],[11,2],[7,6],[11,4],[11,6]

思考：变形题。原来的 $\text{matrix}[i][j]=a[i]+b[j]$

解法：

二分区间：[a[0]+b[0], a[N-1]+b[M-1]]

比较方式：2个指针，p1指向a，p2指向b，每次p1++后，p2往后+1移动指针，直到a[p1]+b[p2]超过target为止，然后往回-1移动直到a[p1]+b[p2]小于target为止。

举一反三

- Leetcode 668
- 题目描述：
 - 给定2个数n,m, 从[1,n],[1,m]中各挑出一个元素, 返回乘积第k小的元素
 - 变形-> $\text{matrix}[i][j]=a[i]*b[j]$
- Leetcode 719
- 题目描述：
 - 给定一个数组, 求数组内两两相减后的绝对值的第k大的数。
 - 变形-> $\text{matrix}[i][j]=a[i]-b[j]$
- Leetcode 786
- 题目描述：
 - 给定一个质数数组, 求数组内两两做差的第k小的数
 - 变形-> $\text{matrix}[i][j]=a[i]/b[j]$

Q&A

分享预告：

- tcmalloc原理
- 无锁队列与ringbuffer
- Go语言并发
- 算法题