

面试题40. 最小的k个数

Friday, July 1, 2022 9:50 AM

<https://leetcode.cn/problems/zui-xiao-de-kge-shu-lcof/>

简单

输入整数数组 arr，找出其中最小的 k 个数。例如，输入4、5、1、6、2、7、3、8这8个数字，则最小的4个数字是1、2、3、4。

示例 1:

输入: arr = [3,2,1], k = 2

输出: [1,2] 或者 [2,1]

思路:

- 一、直接排序取前k个数 $O(n\log n)$ $O(\log n)$
- 二、使用堆排序，大根堆先输入k个数，然后维护这个k大小的堆
每次遍历的新值与根比较，大于根无需处理，小于根把根删除再入堆。（维护k个小的数组成的堆） 每次插入 $O(\log k)$ 最差n次插入 故 $O(n\log k)$,空间 $O(k)$

```
class Solution {
    public int[] getLeastNumbers(int[] arr, int k) {
        int[] res = new int[k];
        if (k == 0) {
            return res;
        }
        // 此处为java定义大根堆结构
        PriorityQueue<Integer> queue = new PriorityQueue<Integer>(new
        Comparator<Integer>() {
            public int compare(Integer num1, Integer num2) {
                return num2 - num1;
            }
        });
        // 先建立k个数构成的堆
        for (int i = 0; i < k; ++i) {
            queue.offer(arr[i]);
        }
        // 之后开始维护大根堆，peek为取根元素，但不删除；poll为删除根元素；offer加入新元素。
        for (int i = k; i < arr.length; ++i) {
            if (queue.peek() > arr[i]) {
                queue.poll();
                queue.offer(arr[i]);
            }
        }
        // 将堆内元素放入结果数组
        for (int i = 0; i < k; ++i) {
            res[i] = queue.poll();
        }
    }
}
```

```
        return res;
    }
}
```