1. 取topK
2. 桶排序+小顶堆堆（只要新来的比堆中最小的要小，就可以替换掉最小的啦）

桶：HashMap

HashMap.put(key , value);

HashMap.remove(key);

HashMap.size();

HashMap.containsKey(key);

HashMap.containsValue(value);

Key\_list = HashMap.keySet();

Value\_List = HashMap.ValueSet();

堆：PriorityQueue

// 无参构造，这个heap就是升序的，意味着就是最小堆

// 自创一个Comparator的话，可以堆不同类型进行比较，同时改成小顶堆，那么(a , b) -> return a - b ，两个顺序不变相减就是小顶堆啦（详见顺序相同就是逆序，最大堆）

Queue<Integer , Integer> heap = new PriorityQueue<>();

Queue<Integer , Integer> heao = new PriorityQueue<>(new Comparator<>(){

@Override

Public int compare(Integer a , Integer b){

Return a - b ;

}

}

);

//访问堆顶元素

PriorityQueue.peek();

//删除堆顶元素，同时返回堆顶元素

PriorityQueue.remove();

PriorityQueue.add(value);