1.排序解决法：

在排序算法中，相对较优的是快速排序，效率为nlog（n），将数组从大到小进行排序，第k大的数则为array[k-1]，快速排序算法在另外总结中，此处不详细介绍。

2.类快排算法：

由于只要求找出第K大的数，没必要将数组中所有值进行排序。

快排中的分治算法，返回key在数组中的位置，如果key的位置正好等于k-1，那么问题则得到解决，如果key的位置不等于k-1，可使用递归查找对应子数组。直到key的位置等于k-1，则找对问题的解。

```
public static int findK(int [] array, int left, int right, int k) {
    int i = partition(array, left, right);
    if(i == k - 1) {
        return array[k-1];
    }else if (i > k - 1) {
        return findK(array, left, i - 1, k);
    }else if (i < k - 1) {
        return findK(array, i + 1, right, k);  //这个地方返回的是下标，所以不需要k-i
    }
    return 0;
}

public static int partition(int[] array, int left, int right) {
    int k = array[left];
    int i = left;
    int j = right;
    while (j > i) {
        while (array[j] < k && j > i) {
            j--;
        }
        if (j > i) {
            array[i] = array[j];
            i++;
        }
        while (array[i] > k && j > i) {
            i++;
        }
        if (j > i) {
            array[j] = array[i];
            j--;
        }
    }
    array[i] = k;
    return i;
}

public static void main(String [] args) {
      int [] array = new int [] {9,7,4,5,2,1,3,8,6,0};
      System.out.println(findK(array, 0, 9, 5));
}
```

3.最小堆解法

最小堆是一种特殊的数组结构，它的实质是一个完全二叉树，且树中子节点的值均小于父节点的值。考虑到只需要找到第k大的数，构造一个大小为k的最小堆，堆中根节点为最小值。如果数组中最大的几个数均在堆中，那么堆中根节点的值就是问题的解。