设计一个方法，删除一个有 N 个元素的数组 A 中所有的重复元素，返回仍留在数组 A 中的元素个数。函数的时间复杂度必须为 O（N log N）（提示：用快速排序作为一个预处理步骤）。试设计一个函数，在一个规模为 N 的无序数组中找出第 k 个大的元素。

（1） 要求时间复杂度为 O（N + klogN））。

（2） 要求时间复杂度为 O（Nlogk）

【解】可以先用快速排序将数组元素进行排序，然后顺序扫描数组，删除重复元素。排序数组元素的时间复杂度是 O(NlogN)，在一个有序数组中删除重复元素的时间复杂度是 O(N)，总的时间复杂度是 O(NlogN)。具体实现见代码清单 10-12。

代码清单 10-12 程序设计题 1 的代码

```
1.  template <class T>
2.  int deleteDuplicate(T a[], int size)
3.  {    int i, j;
4.
5.       quickSort(a, size);
6.       for (i = 1, j = 0; i < size; ++i)
7.            if (a[i] != a[j]) a[++j] = a[i];
8.       return j+1;
9.  }
```

deleteDuplicate 函数首先调用 quickSort 函数对数组 a 进行排序。第 6、7 行完成了了去重复的工作。去重复是通过两个指针来完成。i 顺序扫描整个数组，j 指向不重复元素中的最后一个。第 0 个元素肯定是不重复的，所以开始时 j 指向 0，然后从 1 号单元开始顺序扫描。如果 a[i]等于 a[j]，表示遇到了一个重复元素，该元素不予考虑，于是 i 加 1。如果 a[i]和 a[j]不相等，表示遇到了一个新元素，于是 j 加 1，把 a[i]存入 a[j]。扫描结束后 j 指向不包含重复元素的数组中的最后一个元素。由于 C++的数组从 0 开始编号，所以不重复的元素个数是 j+1，返回这个值。此时，a 数组的前 j+1 个元素就是所要的结果。