算法

Pratice 1

14130130231

吴天成

**Index**

Chapter 1： 3

Chapter 2： 5

Chapter 3： 8

3.1 Implement Quicksort and Randomized Quicksort. 5

3.2 Answer the following questions. 11

Chapter 4： 11

a. Give an algorithm that satisfies criteria 1 and 2 above. 13

b Give an algorithm that satisfies criteria 1 and 3 above. 13

c Give an algorithm that satisfies criteria 2 and 3 above. 13

d Can you use any of your sorting algorithms from parts (a)–(c) as the sorting method used in line 2 of RADIX-SORT, so that RADIX-SORT sorts n records with b-bit keys in O(bn) time? Explain how or why not. 14

e Suppose that the n records have keys in the range from 1 to k. Show how to modify counting sort so that it sorts the records in place in O(n+k) time. You may use O(k) storage outside the input array. Is your algorithm stable? (Hint: How would you do it for k = 3?) 14

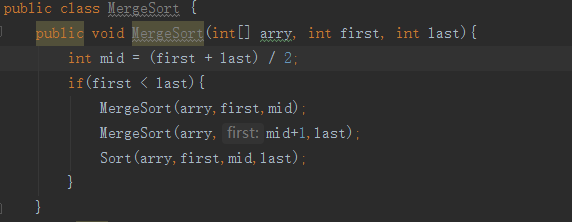
**Chapter 1**

Describe a Θ(n lg n)-time algorithm that, given a set S of n integers and another integer x, determines whether or not there exist two elements in S whose sum is exactly x.

描述一个时间复杂度为nlgn的算法，给出一个含有n个整数的集合和一个整数x，使算法判断该集合是否含有两个元素，使其相加等于x。

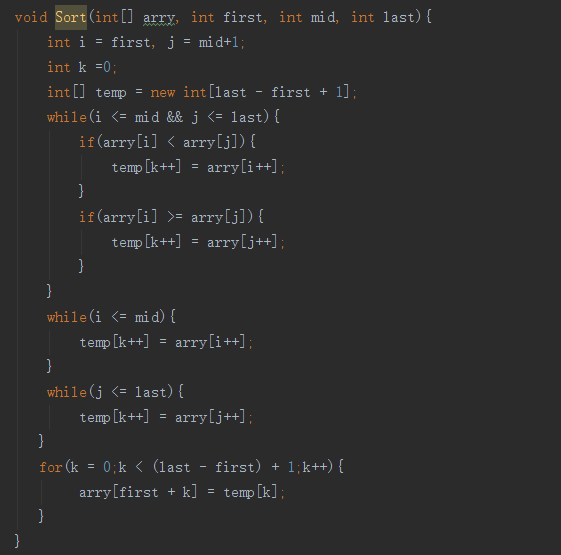
解答：

假设我们先将这个集合排好序，这样判断是否有两个元素相加等于一个数的操作可以分别在集合首尾设立一个指针，若这两个指针之和比所要的数大，则后移尾指针，反之前移前指针。直至两个元素相加刚好等于所要的数或者前指针与尾指针重合。这个操作的时间复杂度为n，再加上之前的排序算法要满足nlgn的时间复杂度的前提。也就是说排序算法的时间负载度为nlgn，归并算法满足条件。所以先进行归并排序再进行求和操作。代码如下：

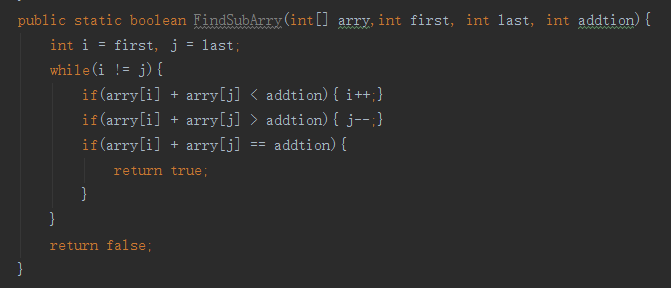


Mergesort先递归的分解集合直至不可再分并使不可再分的数组有序，再进行数组间的合并。

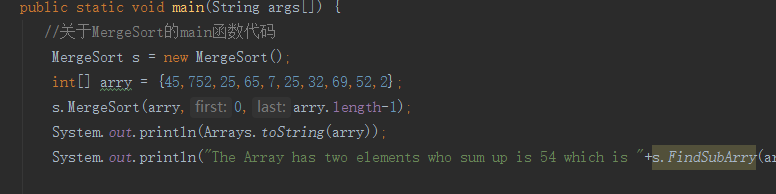
Sort方法 将子数组合并



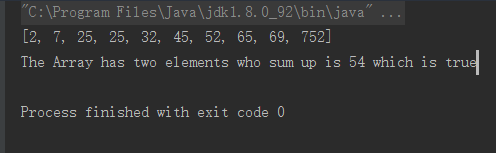
两元素相加操作，返回布尔类型



Main函数调用



结果如下



**Chapter 2**

Implement priority queue.

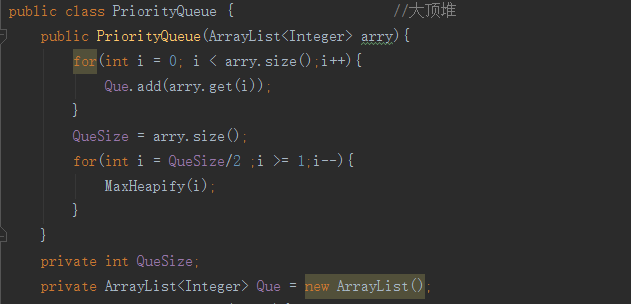
实现优先级队列

解答：

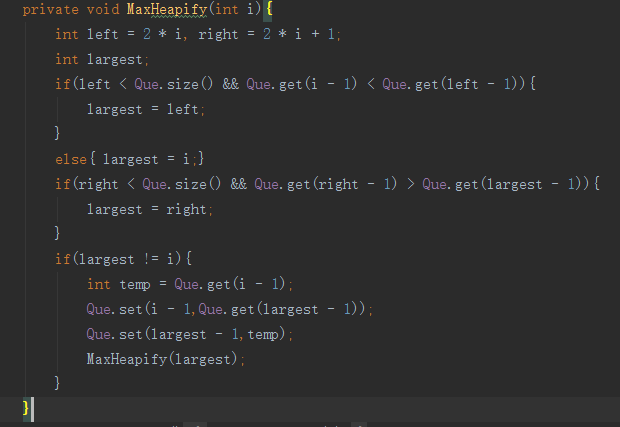
优先级队列

优先级队列（由小到大）是基于大顶堆实现的特殊队列，每个元素都有一个关键字。优先级队列有四个操作（1）MAXIUM （S）查找最大值（2）INSERT（S,x）插入一个新元素（3）EXTRACT-MAX(S)去掉并返回S中具有最大值的元素（4）INCREASE-KEY（S,x,k）将元素x的关键字增加到k。

下图为Priority，类中含有private的一个动态数组Que和一个int常量QueSize。构造函数要求传入一个integer类型的动态数组。方法MaxHeapify是为了维护堆的性质，保证一个节点的左子树右子树均小于该节点。

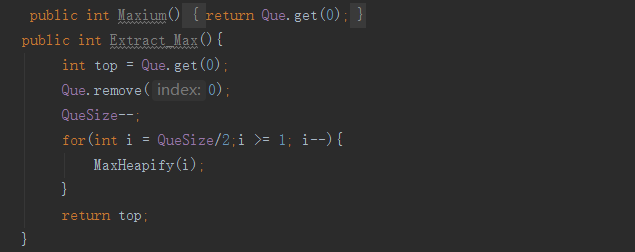


MaxHeapify代码如下



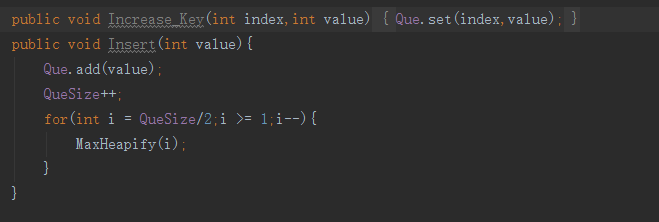
传入一个常量，该常量表示堆的第几个节点，left和right为该节点的左子树和右子树。最后一行进行递归是为了保证若发生了堆内数组交换，交换后的节点仍然满足堆的性质。

Maxium和Extract\_Max代码如下



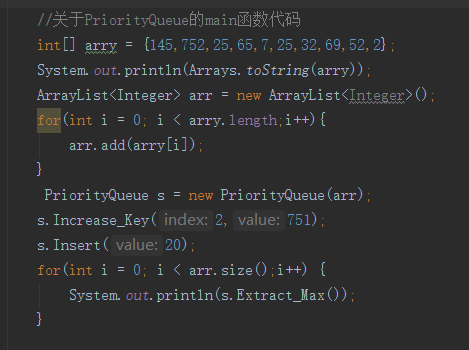
一个大顶堆第一个元素就是其优先级最大的元素，所以get（0），在Exctract\_Max函数中动态数组移除第一个元素后需再次进行堆排序以满足堆的特性。

Increase\_Key和Insert方法代码如下

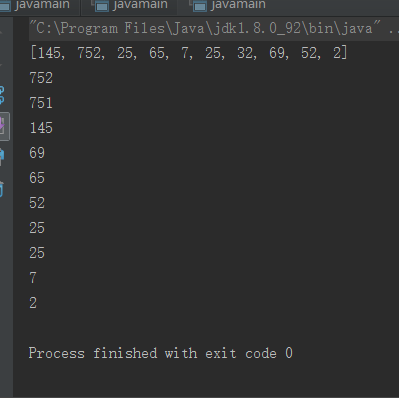


采用动态数组方法实现

Main函数代码



运行结果如下



**Chapter 3**

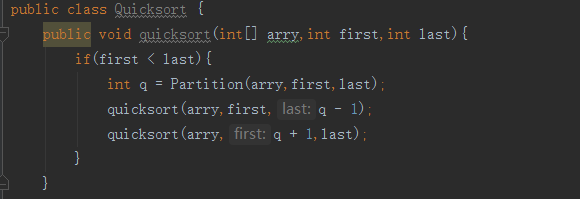
3． Implement Quicksort and Randomized Quicksort. Answer the following questions. (1) How many comparisons will Quicksort do on a list of n elements that all have the same value? (2) What are the maximum and minimum number of comparisons will Quicksort do on a list of n elements, give an instance for maximum and minimum case respectively.

实现快速排序和随机快速排序

解答：

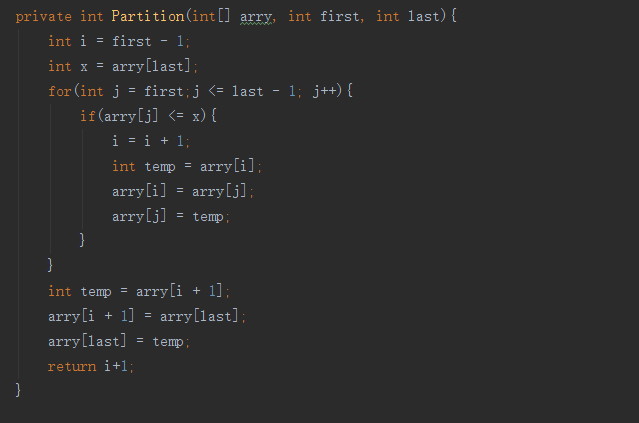
（1）快速排序

快速排序是一种排序方法，通过让元素与某个key值相比较，将元素以该key值为界，大于key值的放在key值的后边，小于key值的放在key值前面。再不断进行递归最终使数列有序。代码如下

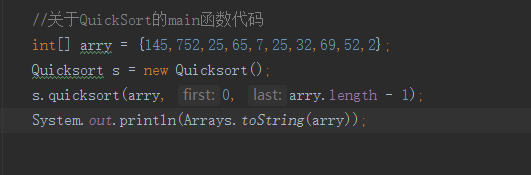


first和last为数组的首尾下标值。我们取数组最后一个值为key值，Partition方法是将元素以key值分解，后边两个quicksort递归是将这两个子数组再进行分界直至数组只剩一个元素。

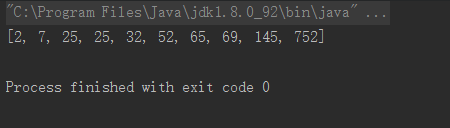
Partition实现代码



Main函数代码



运行结果



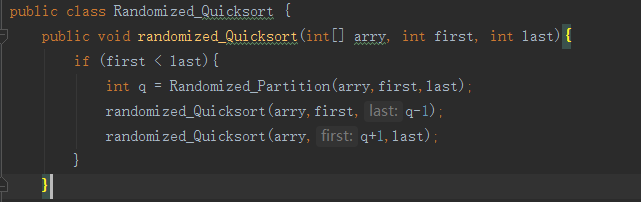
排序成功

（2）随机快速排序

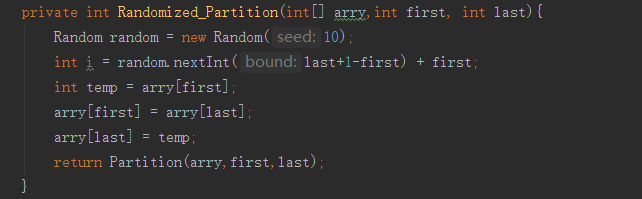
随机快速排序思路与快速排序相似，不同在于选取key值时为了避免所选的数正好为整个数列的最大值这种极端的最坏情况，则取一个随机元素为key值。

代码如下

randomized\_Quicksort与快速排序方法思路相同

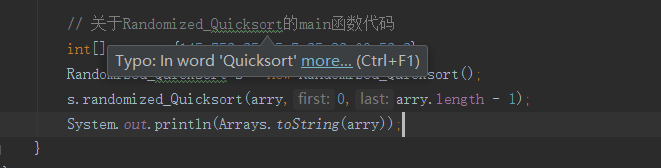


Randomized\_Partition方法

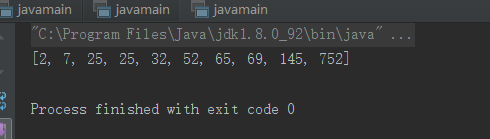


此方法仍需调用快速排序中的Partition方法

Main函数代码



运行结果



（3）How many comparisons will Quicksort do on a list of n elements that all have the same value?

N的平方

What are the maximum and minimum number of comparisons will Quicksort do on a list of n elements, give an instance for maximum and minimum case respectively.

最小比较次数是nlg(n)，e.g每个key值都为数组的中位数，此时要取nlg(n)次。

最大比较次数为n^2，e.g每次选到的值总是最大值或最小值，此时要比较n^2词。

**Chapter 4**

Suppose that we have an array of n data records to sort and that the key of each record has the value 0 or 1. An algorithm for sorting such a set of records might possess some subset of the following three desirable characteristics:

1. The algorithm runs in O(n) time.

2. The algorithm is stable.

3. The algorithm sorts in place, using no more than a constant amount of storage space in addition to the original array.

a. Give an algorithm that satisfies criteria 1 and 2 above.

b. Give an algorithm that satisfies criteria 1 and 3 above.

c. Give an algorithm that satisfies criteria 2 and 3 above.

d. Can you use any of your sorting algorithms from parts (a)–(c) as the sorting method used in line 2 of RADIX-SORT, so that RADIX-SORT sorts n records with b-bit keys in O(bn) time? Explain how or why not.

e. Suppose that the n records have keys in the range from 1 to k. Show how to modify counting sort so that it sorts the records in place in O(n+k) time. You may use O(k) storage outside the input array. Is your algorithm stable? (Hint: How would you do it for k = 3?)

要求对一个0,1数组排序，算法要满足：

1.时间复杂度为n

2.算法稳定

3.算法在空间上为常量级，除了原数组外不能申请额外的空间

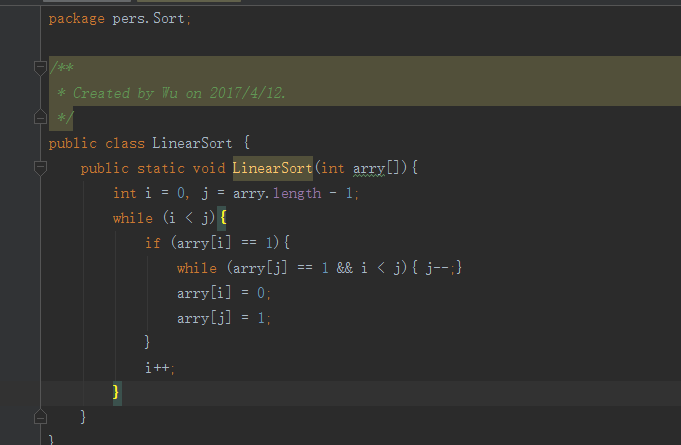
A．给出一个满足1,2条件的算法

B．给出一个满足1,3条件的算法

C．给出一个满足2,3条件的算法

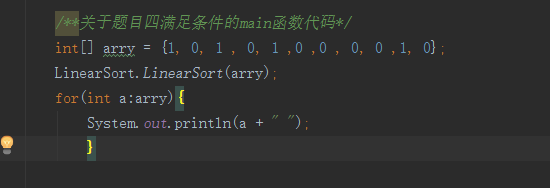
解答：

由于数组只含有0，1，所以我们可以想出如下算法满足以上三个条件

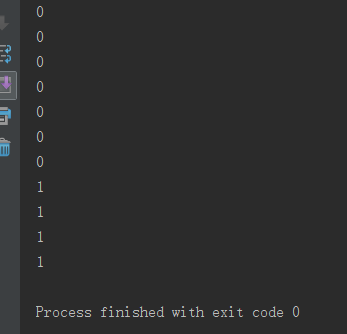


思想为：对于01数组，设置一个头指针和一个尾指针，不断前移头指针，若头指针指向的为1，此时进行尾指针操作，前移尾指针直至尾指针指向0，再将首尾指针值互换，然后继续前移头指针直至头尾指针重合。

Main函数代码

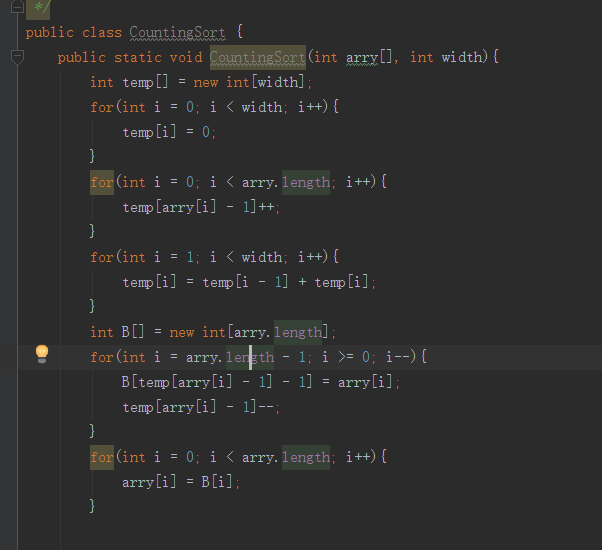


结果



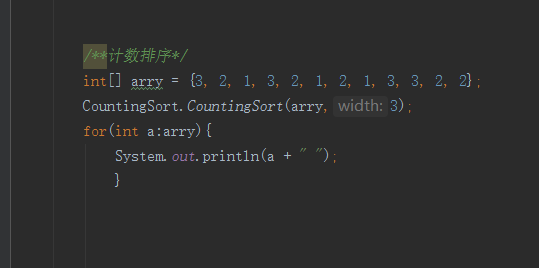
d.运用如上排序算法可以作为基数排序的算法实现基数排序n次，排序算法复杂度为O（b），合起来为O（nb）

e.采用计数排序

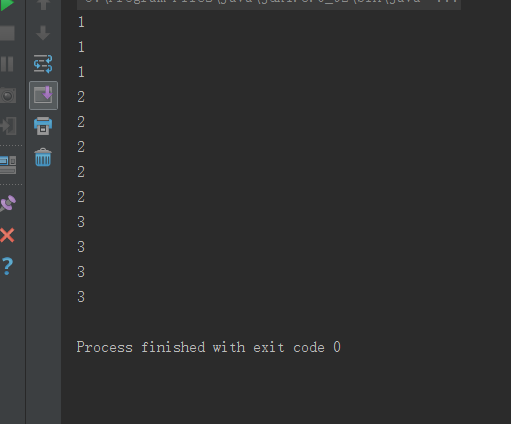


通过额外的k大小的temp数组来存储arry中不同数字的个数，再通过B数组输出

Main函数如下



结果



**源代码在Pratice1/src/pers中，工程目录如下图**

