日报格式:

主 题：日报

汇报人：杨明

日 期：2018年7月30日 星期一

内 容：

1. 今日学习了的内容：
2. 学习了集合中ArrayList和LinkedList的源码分析，比较了二者的优劣
3. 学习了快速排序的优化算法，做了Java基础选择题大概四十道左右
4. 学了简单的JDBC以及SQL注入
5. 学了几个Linux指令
6. 没有解决的问题：
7. 没有吧
8. 明天的学习计划：
9. 继续挖集合源码
10. 学习JDBC工具包的编写
11. 牛客java基础题继续刷

4.学习任务完成度： （90）%[注：0% ~ 100%]

5.对自己今天学习的满意度：（8.5）分 [注：1-10分]

6.对今天的学习做一句话的总结：

总体感觉还好

附录：

算法1：

{

public class QuickSort2 {

/\*\*

\* 以基准元素为准，将数组划分开来，将比基准元素小的放在基准元素左边

\* 比基准元素大的放在基准元素右边，这个过程形象的被称为“挖坑填坑+分治”

\*

\* @param arr 要排序的数组

\* @param left 要排序的数组部分的左边界

\* @param right 要排序数组部分的右边界

\*/

public static void quickSort(int[] arr, int left, int right) {

//递归结束条件

if (arr.length <= 0) {

return;

}

if (left >= right) {

return;

}

//备份边界用来做循环变量

int low = left;

int high = right;

//默认以左边第一个元素为基准元素，将次基准值备份，相当于挖了一个坑

int temp = arr[low];

while (low < high) {

//从后往前遍历，直到找到比基准值小的元素，否则右边界前移

while (low < high && arr[high] >= temp) {

high--;

}

//用找到的比基准值小的这个元素来填充刚才挖出的坑，此时在右边又挖出一个坑，且high指向这个坑

arr[low] = arr[high];

//从前往后遍历，直到找到比基准值大的元素，否则左边界后移

while (low < high && arr[low] <= temp) {

low++;

}

//用找到的比基准值大的这个元素来填充之前挖出的坑，左边又挖出一个坑，且left指向这个坑

arr[high] = arr[low];

}

//循环结束后low指向的位置即为最后基准元素应该在的位置，将其归位

arr[low] = temp;

System.out.println(Arrays.toString(arr));

//循环结束后high和low应该指向同一位置，也就是基准元素所在的位置

//这个位置将数组分为两部分，分别对这两部分进行递归排序

//对左半部分排序

quickSort(arr, left, high - 1);

//对右半部分排序

quickSort(arr, low + 1, right);

}

}

class testQuickSort {

public static void main(String[] arg) {

int[] array = new int[]{3, 2, 5, 8, 4, 7, 6, 9, 1, 10};

System.out.println(Arrays.toString(array));

System.out.println("---------------------------------");

QuickSort2.quickSort(array, 0, 9);

System.out.println("---------------------------------");

System.out.println(Arrays.toString(array));

}

}  
【学习内容真实可靠，给自己的评分和完成度客观真实，不谦虚，不夸大】