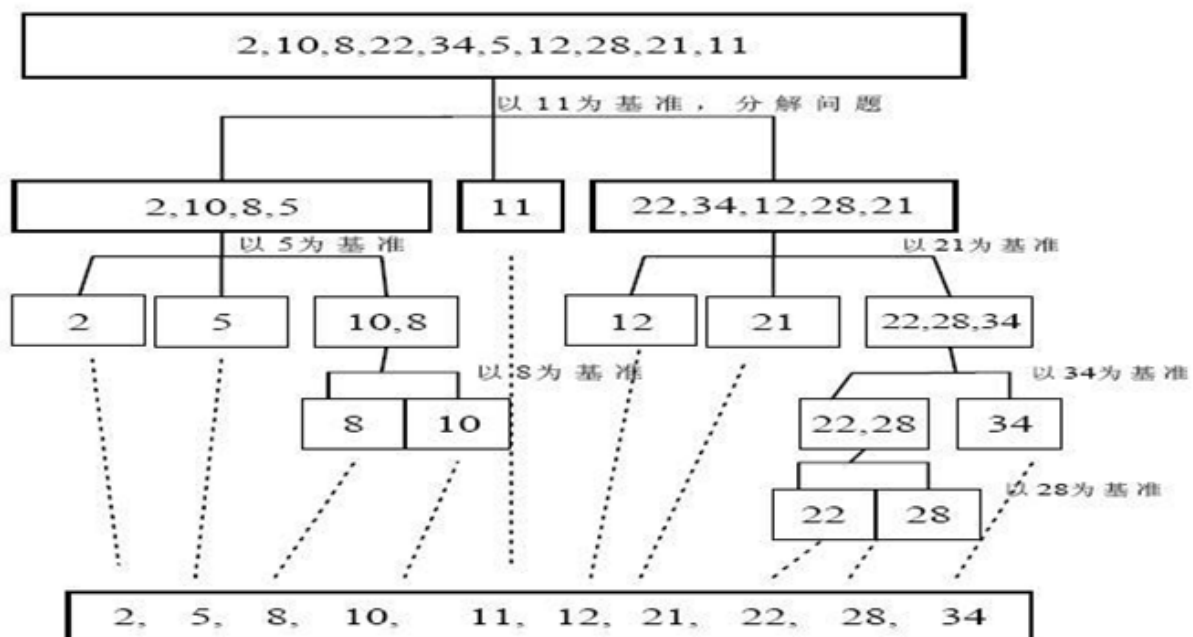


快速排序

1.排序原理:

快速排序（Quicksort）是对冒泡排序的一种改进。**基本思想**是：通过一趟排序将要排序的数据分割成独立的两部分，其中一部分的所有数据都比另外一部分的所有数据都要小，然后再按此方法对这两部分数据分别进行快速排序，整个排序过程可以递归进行，以此达到整个数据变成有序序列。通过**递归**来实现

2.图解



3.代码实现

```
package Sort排序;
```

```
import com.sun.org.apache.xpath.internal.WhitespaceStrippingElementMatcher;
```

```
import java.util.Arrays;
```

```
import java.util.Date;
```

```

import java.util.Random;

/**
 * @author jndeng
 * @create 2019-12-17 19:45
 */
public class QuickSort {
    public static void main(String[] args) {
        int[] array = new int[80000];
        for (int i = 0; i < array.length; i++) {
            array[i] = new Random().nextInt()*80000;
        }
        Date date = new Date();
        long time = date.getTime();

        quickSort(array, 0, array.length - 1);

        Date date1 = new Date();
        long time1 = date1.getTime();
        System.out.println("快排:");
        System.out.println(time1-time);
    }

    public static void quickSort(int[] a, int left, int right) {
        int l = left;
        int r = right;
        int midValue = a[(l + r) / 2];
        int temp=0;
        while (l < r) {
            while (a[l] < midValue) {
                l++;
            }
            while (a[r] > midValue) {
                r--;
            }

            if (l >= r) {
                break;
            }
            temp = a[r];
            a[r] = a[l];
            a[l] = temp;

            //确保循环有出口
            if (a[l] == midValue) {
                r--;
            }
        }
    }
}

```

```

        if (a[r] == midValue) {
            l++;
        }
    }
    //必须有此判断,否则会栈溢出
    if ((l == r)) {
        l++;
        r--;
    }
    //左递归
    if(left<r)
    {
        quickSort(a,left,r);
    }
    //右递归
    if (right > l) {
        quickSort(a,l,right);
    }
}

}

```

几个细节:

1. 在主循环里, 必须有2个判断语句来确保循环能够正常退出, 因为不知道到底是那边先到midValue值
2. 主循环结束后, 递归之前, 必须判断r是否等于l, 对其进行左移和右移. 否则会出现栈溢出
3. 左递归之前需判断left和r的大小
4. 右递归之前需判断right和l的大小