

二分查找、冒泡排序、选择排序、插入排序、快速排序

二分查找

```
package com.it.ui;

public class BinarySearch {
    public static void main(String[] args) {
        int [] array = {1,5,8,11,15,33,35,45,56,77,89};
        int target = 45;

        int idx = binarySearch(array, target);
        System.out.println(idx);
    }

    public static int binarySearch(int[] a, int t){
        int l=0, r=a.length-1, m;

        while(l <= r){
            // 如果是用 m = (l + r) / 2;的话,
            // 如果数值超过整数的最大范围 (21亿多),
            // 然后目标又比左边大的话, (l+r) 就会超出整数的最大范围, 除以2得到的是负数。
            // 这里需要用位运算来解决, 而且运行效率更高。
            m = (l + r) >>> 1;
            if(a[m] == t){
                return m;
            }else if (a[m] < t){
                l = m + 1;
            }else{
                r = m - 1;
            }
        }

        return -1;
    }
}
```

如果是用 $m = (l + r) / 2$;的话, 如果数值超过整数的最大范围 (21亿多), 然后目标又比左边大的话, $(l+r)$ 就会超出整数的最大范围, 除以2得到的是负数。这里需要用位运算来解决, 而且运行效率更高。

排序算法能力要求:

能够手写冒泡、快排的代码 (熟练)

冒泡排序

初步实现:

```
package com.it.ui;

import java.util.Arrays;

public class BubbleSort {
    public static void main(String[] args) {
        int[] a = {5, 9, 7, 4, 1, 3, 2, 8};

        bubble(a);
    }

    public static void bubble(int[] a){
        for(int j = 0; j < a.length - 1; j++){
            // 一轮冒泡
            for (int i = 0; i < a.length - 1; i++){
                if (a[i] > a[i + 1]){
                    swap(a, i, i + 1);
                }
            }
            System.out.println("第"+j+"轮冒泡: " + Arrays.toString(a));
        }
    }

    public static void swap(int[] a, int i, int j){
        int t = a[i];
        a[i] = a[j];
        a[j] = t;
    }
}
```

```
BubbleSort x
"C:\Program Files\Java\jdk1.8.0_341\bin\java.exe" ...
第0轮冒泡: [5, 7, 4, 1, 3, 2, 8, 9]
第1轮冒泡: [5, 4, 1, 3, 2, 7, 8, 9] 这里还要比较一次 (无效比较)
第2轮冒泡: [4, 1, 3, 2, 5, 7, 8, 9]
第3轮冒泡: [1, 3, 2, 4, 5, 7, 8, 9]
第4轮冒泡: [1, 2, 3, 4, 5, 7, 8, 9] 其实到这里已经有序了
第5轮冒泡: [1, 2, 3, 4, 5, 7, 8, 9]
第6轮冒泡: [1, 2, 3, 4, 5, 7, 8, 9]

Process finished with exit code 0
```