**日期：2020/6/19**

**算法与数据结构测试**

1. **实验目的**
2. 掌握Java中几种重要的排序算法。
3. **实验内容**

1、插入排序

编程要求：

本关任务：请在右侧编辑器中填充代码，补全insertSort(int[] arr)函数，实现指定的排序功能，并打印出排序过程（元素之间以空格隔开），函数参数说明如下： arr：待排序的数组

测试说明：

平台会对你编写的代码进行测试：

测试输入：

4,91,51,2,32

预期输出：

4 91 51 2 32

4 51 91 2 32

2 4 51 91 32

2 4 32 51 91

测试输入：

5,1,151,12,22,100

预期输出：

1 5 151 12 22 100

1 5 151 12 22 100

1 5 12 151 22 100

1 5 12 22 151 100

1 5 12 22 100 151

2、选择排序

编程要求：

本关任务：请在右侧编辑器中填充代码，补全selectSort(int[] arr)函数，实现指定的排序功能，并打印出排序过程（元素之间以空格隔开），函数参数说明如下：

arr：待排序的数组

测试说明：

平台会对你编写的代码进行测试：

测试输入：

49,16,23,65,8

预期输出：

8 49 23 65 16

8 16 49 65 23

8 16 23 65 49

8 16 23 49 65

测试输入：

51,15,21,6,108,99

预期输出：

6 51 21 15 108 99

6 15 51 21 108 99

6 15 21 51 108 99

6 15 21 51 108 99

6 15 21 51 99 108

3、冒泡排序

编程要求：

本关任务：请在右侧编辑器中填充代码，补全BubbleSort(int[] arr)函数，实现指定的排序功能，并打印出排序过程（元素之间以空格隔开），函数参数说明如下： arr：待排序的数组

测试说明：

平台会对你编写的代码进行测试：

测试输入：

24,3,12,6

预期输出：

3 12 6 24

3 6 12 24

3 6 12 24

测试输入：

49,97,4,13,27,49

预期输出：

49 4 13 27 49 97

4 13 27 49 49 97

4 13 27 49 49 97

4 13 27 49 49 97

4 13 27 49 49 97

4、折半查找

编程要求：

给定一个 n 个元素有序的（升序）整型数组 nums 和一个目标值T ，写一个函数搜索 nums 中的 T，如果目标值存在返回下标，否则返回 -1。

测试说明：

样例 1:

测试输入: n = 6 ，nums = [-1,0,3,5,9,12]， T = 9

预期输出: 4

解释: 9 出现在 nums 中并且下标为 4

样例 2:

测试输入: n = 6，nums = [-1,0,3,5,9,12], T = 2

预期输出: -1

解释: 2 不存在 nums 中因此返回 -1

5、折半查找

编程要求：

使用二分查找，实现int mySqrt(int x)函数，求平方根的功能。

计算并返回 x的平方根，其中 x 是非负整数。

由于返回类型是整数，结果只保留整数的部分，小数部分将被舍去。

测试说明：

样例1：

测试输入：4

预期输出：2

样例2：

测试输入：8

预期输出：2

解释：8 的平方根是 2.82842..., 由于返回类型是整数，小数部分将被舍去。

6、折半查找

编程要求：

补全右侧函数，实现计算目标值T在数组nums中位置的功能，将计算的结果作为返回值返回。

测试说明：

样例1：

测试输入: n = 7 , nums = [4,5,6,7,0,1,2], T = 0

预期输出: 4

样例2：

测试输入: n = 7 , nums = [4,5,6,7,0,1,2], T = 3

预期输出: -1

1. **实验结果**

1、

public class Sort{

        /\*\*

         \*@param arr 待排序数组

         \*/

          public void insertSort(int[] arr) {

        for(int i=1;i<arr.length;i++){

            int temp = arr[i];

            int j;

            for(j=i-1;j>=0;j--){

                if(arr[j]>temp){

                    arr[j+1]=arr[j];

                }else{

                    break;

                }

            }

            arr[j+1] = temp;

            for(int k=0;k<arr.length;k++){

                System.out.print(arr[k]+" ");

            }

            System.out.println();

        }

          }

  }

2、

public class Sort{

    /\*\*

     \*@param arr 待排序数组

     \*/

    public void selectSort(int[] arr){

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        for(int i=0;i<arr.length-1;i++){

            for(int j=i+1;j<arr.length;j++){

                if(arr[i]>arr[j]){

                    int temp = arr[i];

                    arr[i]=arr[j];

                    arr[j]=temp;

                }

            }

            for(int k=0;k<arr.length;k++){

                System.out.print(arr[k]+" ");

            }

            System.out.println();

        }

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

3、

public class Sort {

    /\*\*

     \*@param arr 待排序数组

     \*/

    public void BubbleSort(int[] arr){

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        int s;

        for(int i=0;i<arr.length-1;i++){

            for(int j=0;j<arr.length-1-i;j++){

                if(arr[j]>arr[j+1]){

                    s=arr[j];

                    arr[j]=arr[j+1];

                    arr[j+1]=s;

                }

            }

            for(int k=0;k<arr.length;k++){

                System.out.print(arr[k]+" ");

            }

            System.out.println();

        }

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

4、

import java.util.Scanner;

public class Task {

        static int[] nums;

    public static void main(String[] args) {

        Scanner sc =new Scanner(System.in);

        int n = sc.nextInt();

        nums = new int[n];

        for(int i = 0 ; i < n ;i++){

            nums[i] = sc.nextInt();

        }

        int T = sc.nextInt();

        int result = search(T);

        System.out.println(result);

    }

    public static int search(int T){

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        int high = nums.length-1;

        int low=0;

        while(low<=high){

            int mid = (low +high)/2;

            if(nums[mid]==T)return mid;

            else if(nums[mid]<T)low=mid+1;

            else high=mid-1;

        }

        return -1;

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

5、

public class Task {

    public int mySqrt(int x){

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        int low=0;

        int high=x;

        while(low <= high){

            long mid=(low+high)/2;

            if(mid\*mid==x)return (int)mid;

            else  if(mid\*mid<x)low=(int )(mid+1);

            else high=(int)(mid-1);

        }

        return high;

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    };

}

6、

public class Task {

    public int search(int n,int[] nums,int T){

        /\*\*\*\*\*\*\*\*\* Begin \*\*\*\*\*\*\*\*\*/

        for(int i=0;i<n;i++){

            if(nums[i]==T)return i;

        }

        return -1;

        /\*\*\*\*\*\*\*\*\* End \*\*\*\*\*\*\*\*\*/

    }

}

1. **实验总结**

通过此次实验，使我对选择算法、折半查找这些基础算法有了更进一步的了解，并通过实验平台实践这些算法。以后需要多上机练习才能更好的掌握这门语言。