

## 实验 2.1 最小差值

计科 1903 201914020128 陈旭

### 一. 问题分析

#### 1). 问题与功能描述

1.需要处理的数据是一串整型(int 型)数字。

2.实现的功能有：存储这 n 个数；

将这 n 个数进行排序；

对比每个数和他下一个数的差值的绝对值；

将这些差值的绝对值进行对比；

输出最小的插值的绝对值。

3.直接采用标准输出整数即可。

#### 2). 样例分析

1. 求解方法：① 将线性表中元素按照顺序排好，

② 对比每一个元素和它相邻的元素的差值绝对值，用一个数记录最小值，如果新的绝对值比前面的值要小，则记录的数被赋值为新的最小绝对值。最终输出这个用来记录最小值的数。

2. 样例求解：① 【样例一】

输入的数有 5 个，为 1 5 4 8 20 将这五个数全部尾插入线性表，然后进行排升序，得到 1 4 5 8 20；然后从 1 到 4 对比差的绝对值，为 3，1，3，12。显然最小差值为 1。最后输出 1。

② 【样例二】

输入的数据个数仍为 5，五个数分别为 9 3 6 1 3；尾插之后排序，得到 1 3 3 6 9。计算每两个相邻数的差值为 2 0 3 3。显然最小差值为 1，最终输出 1。

### 二. 数据结构分析

1) 数据对象 本题处理的数据为整型(int)。

2) 数据关系 本题处理的数据为整型有限数据，只需要排序并且比较，无需大规模检索，为线性结构，故本题只需要使用线性表这一数据结构即可。

### 3) 基本操作

- **【功能描述】** 将元素尾插入线性表  
  
**【名字】**        append  
  
**【输入】**        本题中为整型数字  
  
**【输出】**        无
- **【功能描述】** 将元素排升序  
  
**【名字】**        setorder  
  
**【输入】**        无  
  
**【输出】**        无
- **【功能描述】** 将当前元素的位置移动到线性表开头  
  
**【名字】**        movestart  
  
**【输入】**        无  
  
**【输出】**        无
- **【功能描述】** 输出线性表的当前长度  
  
**【名字】**        length  
  
**【输入】**        无  
  
**【输出】**        一个整数，即线性表长度
- **【功能描述】** 返回当前元素值  
  
**【名字】**        getvalue  
  
**【输入】**        无  
  
**【输出】**        本题为整数

- **【功能描述】** 将当前元素的位置向后移动一位

**【名字】**      next

**【输入】**      无

**【输出】**      无

#### 4) 物理实现

```
void append(const E& it)
{
    assert(listsize<maxsize)
    array[listsize]=it;
    listsize++;
} //尾插一个函数
void setorder ()
{
    sort(array, array+listsize);
} //排升序
void movetostart()
{
    curr=0;
} //将当前元素的位置移动到线性表开头
int length() const
{
    return listsize;
} //输出线性表的当前长度
void next()
{
    assert(curr<listsize);
    curr++;
} //返回当前元素值
const E& getvalue()
{
    assert((curr>=0)&&(curr<=listsize));
    return array[curr];
} //将当前元素的位置向后移动一位
```

### 三. 算法分析

- **算法思想：**先排序，排序之后只需要考虑相邻元素的差的绝对值。实现起来会更加

简便

- **关键功能的算法步骤:**

```
append(num); //插入数据。
for (int i=0; i<vec_num.length()-1; i++)
{
    if (i==0)
    {
        temp1=getvalue();
        next();
        temp=abs(getvalue()-temp1); //如果是第一次判断，将中间
                                   变量赋值给第一个
                                   元素和第二个元素
                                   的差
    }
    else
    {
        temp1= getvalue();
        next();
        if (abs(getvalue()-temp1)<temp)
            temp=abs(getvalue()-temp1);
    } //其他情况判断是否比初始情况或者上一个情况更小，如果是，记录那
      个更小的。
}
cout << temp << endl; //输出得到的最小值。
```

- **性能分析**

【空间复杂度】每输入一个值便进行一次插入操作，进行一次判断选择性记录数据，得空间复杂度为  $\Theta(n)$ 。

【时间复杂度】先循环输入一个值，时间开销为  $C_1n$ ，再进行排序,开销为  $C_2n\log n$ ，然后循环判断,开销为  $C_3n$ ，综上时间复杂度为  $O(n)$ 。