

week7实验：C程序测试与断点调试(冒泡排序/插入排序/选择排序)

15331302 王翔宇 软工2015级1班 教务一班

MarshallW906@GitHub

week7实验：C程序测试与断点调试(冒泡排序/插入排序/选择排序)

Overview

对debugging的看法

使用工具：

三种排序简介

1.1冒泡排序

1.2插入排序

1.3选择排序

Assignments

Assignment 0

Assignment 1

Question 2：测例

Question 1 & 3

冒泡排序

插入排序

选择排序

Question 4：Pseudocode

冒泡排序

插入排序

选择排序

Discussion , Cooperation & Summary

option 3：Helping Others

option 4：Personal Summary

总结

Overview

学习基本的C程序测试/调试方法，养成系统的debug方法。

debug实例为三种时间复杂度为 $O(n)$ 的初级排序算法，即与此同时学习该三种排序。

对debugging的看法

平均来说，在发布程序之前，debug用的时间通常是初次编写用时的六倍。我们处于入门的阶段，写程序简单的问题便经常出错，此时用普通肉眼扫描逐行检查的方法尚可；但当问题和代码的规模同步扩大时，则需要一个系统的方法。断点调试即是一个好方法。

使用工具：

1. sublime Text 3 ->文本编辑
2. Dev C++ 5.11 ->断点调试
3. cmd
4. 马克飞象 ->实验报告书写

三种排序简介

1.1冒泡排序

重复地走访过要排序的数列，一次比较两个元素，如果他们的顺序错误就把他们交换过来。走访数列的工作是重复地进行直到没有再需要交换，也就是说该数列已经排序完成。这个算法的名字由来是因为越小的元素会经由交换慢慢“浮”到数列的顶端。

1.2插入排序

通过构建有序序列，对于未排序数据，在已排序序列中从后向前扫描，找到相应位置并插入。

1.3选择排序

首先在未排序序列中找到最小（大）元素，存放到排序序列的起始位置，然后，再从剩余未排序元素中继续寻找最小（大）元素，然后放到已排序序列的末尾。以此类推，直到所有元素均排序完毕。

Assignments

Assignment 0

: 命令行学习。

```
F:\>cd F:\rustbin\ProgDesignI\week7

F:\rustbin\ProgDesignI\week7>gcc 1005.c -o 1005.exe

F:\rustbin\ProgDesignI\week7>1005.exe
2005 6 3
March 2005
    Sun    Mon    Tue    Wed    Thu    Fri    Sat
         1     2     3     4     5
    6     7     8     9    10    11    12
   13    14    15    16    17    18    19
   20    21    22    23    24    25    26
   27    28    29    30    31

F:\rustbin\ProgDesignI\week7>1005.exe < input.txt
March 2005
    Sun    Mon    Tue    Wed    Thu    Fri    Sat
         1     2     3     4     5
    6     7     8     9    10    11    12
   13    14    15    16    17    18    19
   20    21    22    23    24    25    26
   27    28    29    30    31

F:\rustbin\ProgDesignI\week7>_
```

cmd是一个非常有用的东西。

当Path配置好了之后，我还是习惯任何事件的产生和中止都在bash下运行。

Assignment 1

Question 2 : 测例

~对以下三个程序，所有的测例使用两种结合：

1. 题目给出的随机测例
2. 该类排序最坏情况下的测例（不同长度）

input:

6

3 1 3 2

5 1 2 3 4 5

4 99 100 22 33

2 2 1

3 3 2 1

5 5 4 3 2 1

output:

1 2 3

1 2 3 4 5

22 33 99 100

1 2

1 2 3

1 2 3 4 5

Question 1 & 3

冒泡排序

源代码从直观看到很多错误：

1. 开头定义很多变量但没有使用
2. 存在多次test，但代码只实现了单次test
3. 变量交换写错
4. 对冒泡排序操作边界理解错误（只需要到n-1）即可
5. 代码风格
6. 代码风格问题，cpplint.py测试截图如下：

```
F:\rustbin\ProgDesignI\week7shiyang>cpplint.py bubble_ori.c
bubble_ori.c:0: No copyright message found. You should have a line: "Copyright
[year] <Copyright Owner>" [legal/copyright] [5]
bubble_ori.c:18: Missing space before ( in if( [whitespace/parens] [5]
Done processing bubble_ori.c
Total errors found: 2
```

可以看出，问题有如下几点：

~无版权声明。

~if之后的括号前应该有空格

~PS：没有缩进（cpplint.py未查出）

插入排序

源代码的错误：

1. 同样只实现了对单次test的处理
2. 变量交换和temp的重置都存在问题。
3. 代码风格

cpplint.py测试截图如下：

```
F:\rustbin\ProgDesignI\week7shiyang>cpplint.py insertion_ori.c
insertion_ori.c:0: No copyright message found. You should have a line: "Copyright
[year] <Copyright Owner>" [legal/copyright] [5]
insertion_ori.c:8: Missing space before ( in for( [whitespace/parens] [5]
insertion_ori.c:10: Missing spaces around = [whitespace/operators] [4]
insertion_ori.c:10: Missing spaces around < [whitespace/operators] [3]
insertion_ori.c:10: Missing space before ( in for( [whitespace/parens] [5]
insertion_ori.c:10: Missing space before { [whitespace/braces] [5]
insertion_ori.c:12: Missing spaces around = [whitespace/operators] [4]
insertion_ori.c:12: Missing spaces around > [whitespace/operators] [3]
insertion_ori.c:12: Missing space before ( in for( [whitespace/parens] [5]
insertion_ori.c:12: Missing space after ; [whitespace/semicolon] [3]
insertion_ori.c:12: Missing spaces around && [whitespace/operators] [3]
insertion_ori.c:13: Missing spaces around = [whitespace/operators] [4]
insertion_ori.c:14: Missing space before ( in for( [whitespace/parens] [5]
Done processing insertion_ori.c
Total errors found: 13
```

问题：

- ~无版权声明。
- ~其余全部是无空格的问题
- ~缩进 (cpplint.py 同未查出。)

选择排序

源代码的错误：

1. 编译错误：min未声明、括号不完整
2. 仅处理单组测例
3. min的用途不明：方便的做法是将min用作最小值的索引值
4. 交换语句位置错误：应在外部循环中——选择排序最多只有N次交换。
5. 代码风格

cpplint.py测试截图：

```
F:\rustbin\ProgDesignI\week7shiyang>cpplint.py selection_ori.c
selection_ori.c:0:  No copyright message found.  You should have a line: "Copyri
ght [year] <Copyright Owner>"  [legal/copyright] [5]
selection_ori.c:8:  Missing space before ( in for(  [whitespace/parens] [5]
selection_ori.c:10:  Missing spaces around =  [whitespace/operators] [4]
selection_ori.c:10:  Missing space before ( in for(  [whitespace/parens] [5]
selection_ori.c:10:  Missing space before {  [whitespace/braces] [5]
selection_ori.c:12:  Missing space before ( in for(  [whitespace/parens] [5]
selection_ori.c:13:  Missing spaces around >  [whitespace/operators] [3]
selection_ori.c:13:  Missing space before ( in if(  [whitespace/parens] [5]
selection_ori.c:17:  Missing space before ( in for(  [whitespace/parens] [5]
Done processing selection_ori.c
Total errors found: 9
```

问题：

- ~版权声明
- ~缺少空格
- ~缩进

Question 4 : Pseudocode

冒泡排序

```

begin

int num[10000];
int n, i, j, m;

input n;
loop 1 to n Step 1 {
    input m;
    All of num[0..9999] := 0;
    for (i = 0; i < m; i++) {
        input num[i];
    }

    for (i = 0; i < m - 1; i++) {
        for (j = 0; j < m - i - 1; j++) {
            if (num[j] > num[j + 1])
                swap(num[j], num[j + 1]);
        }
    }
    :output all of the m numbers from the 1st to the last
}

end

```

插入排序

```

begin

int num[10000]
int n, i, j, m, temp

input n

loop 1 to n Step 1 {
    input m

    All of num[0..9999] := 0

    for (i = 0; i < m; i++)
        input num[i];
    for (i = 0; i < m; i++) {
        temp := num[i]
        for (j = i - 1; j >= 0 && num[j] > temp; j--) {
            num[j + 1] := num[j];
        }
        num[j + 1] := temp
    }
    :output all of the m numbers from the 1st to the last
}

end

```

选择排序

```

begin

int num[10000]
int n, m, i, j, min, temp

input n
loop 1 to n Step 1 {
    input m
    All of num[0..9999] := 0
    for (i := 0; i < m; i++)
        input num[i];
    for (i := 0; i < m; i++) {
        min := i
        for (j := i + 1; j < m; j++)
            if (num[min] > num[j])
                min := j;
        swap(num[i], num[min]);
    }

    :output all of the m numbers from the 1st to the last
}

end

```

Discussion , Cooperation & Summary

option 3 : Helping Others

(代码之外的)

帮助四个同学搞定了虚拟机【镜像也是我提供的】

帮助配置了gcc编译器

(代码的)

wk7的西西里题中的那道日历，许多人一开始在疯狂用switch，直接的结果就是代码行数非常多，可读性差、debug难度高。

后来我告诉他们可以用const定义数组，之后用下表引用的方式，效果立竿见影。

一些同学不能很好的归纳其中的规律，造成代码非常繁琐，也是很不好。那道题我认为要用好 %7。把自己的思路分享之后，问我的同学也是觉得帮助不小。

option 4 : Personal Summary

做完这个实验报告是在期中考结束之后。期中考上机AK，笔试不至于太差，从短期来看还是比较满意。

但原因是TA放了全水，因此尝到了一点甜头。目测今后的每一次测验都会越来越难，相应地，我也要多吃一些题，多学一些算法，否则结果将很惨淡。

总结

这次报告应该可以让所有同学对三种初级排序算法有了比较好的了解，而且debug和伪代码的书写过程会使慢慢感受到程序设计真正的意义和乐趣所在。