

Beyond

Interview-Amazon-OA1-Debug

📅 2017-09-06 | 📁 Interview

Debug 题只有七题。

总体原则(引用小土豆博客) <http://wdxutub.com/interview/14520850399861.html>

- 1 OA1 - Debugging
- 2
- 3 一些可能的问题:
- 4
- 5 while循环缺少i++造成死循环
- 6 Print Pattern: for-loop里一共两句话但是没有用大括号, 所以第二句没有被包含进去
- 7 insert sort descending order: <> 反了
- 8 selection sort: arr[min]>arr[x] 改成 arr[y].
- 9 reverse array: arr[len-1] 改成 arr[len-i-1], 循环结束前去掉 len+=1;
- 10 循环里要加上 i++, 否则死循环
- 11 曼切斯特如果arr[i-1], arr相等为0, 否则为1, 要注意不但==要改成!=, ret[0]也要加一下, 不然有一个case
- 12 解题技巧
- 13
- 14 因为地里面经分散, 大家又说很简单, 多办懒得附上题目, 在这里提供几个思路给大家:
- 15
- 16 排序类: 这种题Compile & Run出来的结果, 多半是Sort的顺序反了, 稍微看一下找到关键的if statement把他
- 17 TLE类: 这种结果TLE的多半是while死循环了, 有while的检查一下。
- 18 for loop类: 有for loop的, 检查一下大括号有没有加。
- 19 其他类: 其他特别的bug, 多半地里找一下都有, 记一下就行。

1. ArraySum: 计算一个整数数组的和

bug: sum = arr[i], should be sum += arr[i]

2. CountOccurence: 统计某数在整数数组中出现的次数

bug: while loop i++

3. CheckGrade: 根据百分制分数计算 GPA

bug: 判断条件 “||” should be “&&”

4. RemoveElement: 删除数组中特定位置元素, 超出范围则返回原数组

bug:第七行arr[i++] should be arr[i+1]

```

1  // You can print the values to stdout for debugging
2  public class ShortArray {
3      public static int[] removeElement(int arr[], int index) {
4          int i, j, len = arr.length;
5          if (index < len) {
6              for (i = index; i < len - 1; i++) {
7                  arr[i] = arr[i++]; ====改成==> arr[i] = arr[i+1]
8              }
9              int rarr[] = new int[len - 1];
10             for (i = 0; i < len - 1; i++) {
11                 rarr[i] = arr[i];
12             }
13             return rarr;
14         } else
15             return array;
16     }
17 }
```

5. DigitCount: 将一个整数除以它的位数取余数 num%len(num) == 0 or != 0

bug:错在最后返回的除法时保留原数就行 把num用另一个变量存起来就好 因为num最后变成了0

```

1  public class Digits {
2      public int find(int num) {
3          int count = 0;
4          // 加一行
5          int number = num;
6          //
7          while(num != 0) {
8              num = num/10;
9              count++;
10         }
11         return num % count; =改成=> return number % count;
12     }
13 }
```

6. Sort Array 冒泡排序: < 改 > 或者 > 改 <

```

1  public static int[] sortArray(int[] arr) {
2      int len = arr.length;
3      int small, pos, i, j, temp;
4      for (i = 0; i <= len - 1; i++) {
```

```
5         for (j = i; j < len; j++) {
6             temp = 0;
7             if (arr[i] > arr[j]) { // > 改 <
8                 temp = arr[i];
9                 arr[i] = arr[j];
10                arr[j] = temp;
11            }
12        }
13    }
14    return arr;
15 }
```

7. Sort Array 2 : 选择排序 < 改 > 或者 > 改 <

```
1  public static int[] sortArray2(int arr[]) {
2      int i, max, location, j, temp, len = arr.length;
3      for (i = 0; i < len; i++) {
4          max = arr[i];
5          location = i;
6          for (j = i; j < len; j++) {
7              if (max > arr[j]) { // > 改 <
8                  max = arr[j];
9                  location = j;
10             }
11         }
12         temp = arr[i];
13         arr[i] = arr[location];
14         arr[location] = temp;
15     }
16     return arr;
17 }
```

8. Selection sort 这题告诉我们排序问题也可能是下标出了问题。

bug: if 里面的判断它写的是 arr[min]> arr[x], 改成 arr[min]> arr[y]

```
1  for (x = 0; x < n; x++) {
2      int index_of_min = x;
3      for (y = x; y < n; y++) {
4          if (arr[index_of_min] > arr[x]) {
5              y = index_of_min;
6          }
7          int temp = arr[x];
8          arr[x] = arr[index_of_min];
9          arr[index_of_min] = temp;
10     } }
```

```

11 // selection sort in ascending order 升序
12 for (int x = 0; x < n; x++) {
13     int index_of_min = x;
14     for (int y = x; y < n; y++) {
15         if (arr[index_of_min] > arr[y]) {
16             index_of_min = y;
17         }
18     }
19     int temp = arr[x];
20     arr[x] = arr[index_of_min];
21     arr[index_of_min] = temp;
22 }

```

9. bubble sort < 改 > 或者 > 改 <

```

1 // bubble sort in descending order 降序
2 for (int x = 0; x < n; x++) {
3     for (int y = x; y < n; y++) {
4         if (arr[x] < arr[y]) { // < 改成 > 或者 > 改成 <
5             int temp = arr[x];
6             arr[x] = arr[y];
7             arr[y] = temp;
8         }
9     } }
10
11 // 死循环，不懂面经中这段代码是什么意思。将选择排序和冒泡排序混在一起。
12 for (x = 0; x < n; x++) {
13     int index_of_min = x;
14     for (y = x; y < n; y++) {
15         if (arr[index_of_min] <= arr[x]) {
16             y = index_of_min;
17             int temp = arr[x];
18             arr[x] = arr[index_of_min];
19             arr[index_of_min] = temp;
20         }
21     }
22 }

```

10. insertion sort < 改 > 或者 > 改 <

```

1 for (int i = 1; i < n; i++) {
2     if (arr[i - 1] > arr[i]) {
3         int temp = arr[i];
4         int j = i;
5         while (j > 0 && arr[j - 1] > temp) {
6             arr[j] = arr[j - 1];

```

```

7         j--;
8     }
9     arr[j] = temp;
10 }
11 }

```

11. replace values 数组长度如果是偶数就把所有元素都改成1，奇数就都改成0

bug: for loop 里的 $i \leq \text{len}$, $j \leq \text{len}$ 把 \leq 改成 $<$

```

1  public static int[] replaceValues(int arr[]) {
2      int i, j, len = arr.length;
3      if (len % 2 == 0) {
4          for (i = 0; i <= len; i++) { ==> for (i = 0; i < len; i++) {
5              arr[i] = 0;
6          }
7      } else {
8          for (j = 0; j <= len; j++) { ==> for (j = 0; j < len; j++) {
9              arr[j] = 1;
10         }
11     }
12     return arr;
13 }

```

12. reverse array

但class 的名字居然是 sort array

bug: $\text{arr}[\text{len}-1]$ 改成 $\text{arr}[\text{len}-i-1]$ for 循环里的 $\text{len} += 1$ 去掉

```

1  public static int[] reverseArray(int[] arr) {
2      int i, temp, originalLen = arr.length;
3      int len = originalLen;
4      for (i = 0; i < originalLen / 2; i++) {
5          temp = arr[len - 1]; ==> temp = arr[len - i - 1];
6          arr[len - i - 1] = arr[i];
7          arr[i] = temp;
8          len += 1; ==> //remove len += 1;
9      }
10     return arr;
11 }

```

13. EvenOdd pattern

for 循环缺大括号

```

1  public static void print4(int num) {
2      int i, print = 0;

```

```

3      if (num % 2 == 0) {
4          print = 0;
5          for (i = 0; i < num; i++) ==> for ( i = 0; i < num; i++) {
6              System.out.print(print + " ");
7              print += 2;
8          }
9      } else {
10         print = 1;
11         for (i = 0; i < num; i++) ==> for ( i = 0; i < num; i++) {
12             System.out.print(print + " ");
13             print += 2;
14             ==> }
15     }
16 }

```

14. Manchester:输入一个01数组，输出manchester编码，假设第一个 elem前的数字是 0

bug:里面就一个判断语句，把判断语句的 == 改成 != (考虑 if (i == 0) ?)

result = (A[i] == A[i - 1]); // bug

output[i] = result ? 1 : 0

15. PrintPattern2

a
ab
abc

bug: 错在它把char ch = 'a' 定义放在了for loop 前，移进来就好，同时删掉了没用的变量。

```

1  public static void print2(int row) {
2      for (int j = 0; j <=i; j++) {
3          char ch = 'a';
4          char print = ch;
5          System.out.print((ch++));
6      }
7      System.out.println("");
8  }

```

改为

```

1  public static void print2(int row) {
2      char ch = 'a';
3      char print = ch;
4      for (int j = 0; j <=i; j++) {
5          System.out.print((print++));
6      }
7      System.out.println("");

```

1. PrintPattern3

有一道题是要给一个 row number, 输出相应的行数的pattern。比如:

row number = 3 则, 输出

11

1111

111111

row number = 2 则输出:

11

1111

bug: 第一层 for 循环少了大括号, 导致最后那个System.out.println() 最后只执行一次

```
public static void print3(int row) {  
    int x = 1;  
    for (int i = 1; i <= row; i++) {  
        for (int j = i; j > 0; j--) ==> for (int j = i; j > 0; j--) {  
            System.out.print(x + "" + x);  
            ==> }  
        System.out.println();  
    }  
}
```

1. remove duplicates form unsorted array 循环下标要从 i + 1 开始

2. Array 奇数偶数 for loop i+=2 改成 i++

我在第三题卡了很久 题目是给一个array 要把所有不重複的element相加 最后return sum
function一开始先把array[0]加进了sum才使用Arrays.sort(), 颠倒了顺序 将Arrays.sort()移至前一行就行了. 1point3acres
其他印象比较深刻的是一题关于leap year的 input 是 year, month 要return那指定的年份中那个月份有几天
function内有很多switch 及case 问题是出在2月的部分, 将if statement 改为if(year%4==0&&year%100==0&&year%400!=0) 就行了

distinctNumber == 改成 !=
sumDistinct 把排序移到最前
eliminateVowel default case 不需要i++
checkPalindrome 改成 result==tem
median 把i改成i-size
countDays 建议上网看看闰年的定义
elementRange 把&&改成|

debug里面有一道求一个array里面出现k次的数, 如果有多个则返回最大一个。bug是最后一项处理错了。有点紧张没搞对
还有一个是求子函数求digit sum的时候, bug先把number除了10才求的余。

© 2019  Wayne

Powered by [Hexo](#) | Theme — [NexT.Mist](#) v5.1.4