

# 2019CCF 非专业级别软件能力认证第一轮 (CSP-J) 入门级 C++语言试题阅读题解析

郑德强

(福建省泉州市第七中学 泉州 362000)

**摘要** CSP-J/S 是 CCF 创办的 CSP (软件能力认证) 中面向非专业级的软件能力认证, 于 2019 年首次开设, 分为 CSP-J (入门级, Junior) 和 CSP-S (提高级, Senior) 两组, 分别进行两轮认证, 涉及算法和编程。本文题目来源于 2019 年 10 月 19 日 CCF 首次举办的 (CSP-J) 入门级认证第一轮试题, 对三道阅读题的解题思路及算法设计做了较为详细的描述和解析。

**关键词** 题解; 分析; 算法; 代码

中图法分类号 TP31 TP31 DOI:10.16707/j.cnki.fjpc.2019.z1.001

## Solutions to Reading Program of CSP-J 2019

Zheng deqiang

(Quanzhou No.7 middle School, Quanzhou 362000)

### 1 问题描述

```
1 #include <cstdio>
2 #include <cstring>
3 using namespace std;
4 char st[100];
5 int main() {
6     scanf("%s", st);
7     int n = strlen(st);
8     for (int i = 1; i <= n; ++i) {
9         if (n % i == 0) {
10            char c = st[i - 1];
11            if (c >= 'a')
12                st[i - 1] = c - 'a' + 'A';
13        }
14    }
```

```
15     printf("%s", st);
16     return 0;
17 }
```

### • 判断题

- 1) 输入的字符串只能由小写字母或大写字母组成。 ( )
- 2) 若将第 8 行的 “i=1” 改为 “i=0”, 程序运行时会发生错误 ( )
- 3) 若将第 8 行的 “i<=n” 改为 “i\*i<=n”, 程序运行结果不会改变 ( )
- 4) 若输入的字符串全部由大写字母组成, 那么输出的字符串就跟输入的字符串一样。 ( )

### • 选择题

- 5) 若输入的字符串长度为 18, 那么输入的字符串跟输出的字符串相比至多有 ( ) 个字符不同。

- A. 18      B. 6      C. 10      D. 1  
 6) 若输入的字符串长度为 ( ), 那么输入的字符串跟输出的字符串相比, 至多有 36 个字符不同。  
 A. 36      B. 100000      C. 1      D. 128

## 2 考察的知识点

循环语句及 if 语句的理解, 字符数组的存储和遍历, ASCII 编码, 约数个数定理的运用。

## 3 解题思路

该程序先读入字符串, 接着遍历字符串, 若  $i$  是  $n$  的约数且  $st[i-1]$  的 ASCII 码大于等于 97, 则将  $st[i-1]$  变为  $st[i-1]-32$  的字符 (‘A’ 的 ASCII 码为 65, ‘a’ 的 ASCII 码为 97), 最后将修改后的字符串输出。即程序可实现将字符串中下标为字符串长度约数的小写字母改为大写字母。

### • 判断题

- 1) 考察选手对程序的理解, 该程序只是对于字符进行操作, 并没有将字符限制于大小写字母, 还可能是数字和符号, 只要字符不为空格换行都是符合题意的, 故答案为  $\times$ 。
- 2) 若改为 “ $i = 0$ ”, 则程序运行到第 9 行 “ $n \% i == 0$ ” 时出现整数被零除的情况, 运行时发生错误, 故答案为  $\checkmark$ 。
- 3) 第 8 行为 “ $i <= n$ ”, 则该循环遍历  $[1, n]$  中的所有整数, 若将第 8 行的 “ $i <= n$ ” 改为 “ $i * i <= n$ ”, 则循环遍历  $[1, \sqrt{n}]$  中的整数, 显然后一个范围较小, 程序运行结果可能不同, 故答案为  $\times$ 。
- 4) 若输入的字符串全部由大写字母组成, 则运行过程中所有字符的 ASCII 码均小于 97, 所以字符串不变, 原样输出, 故答案为  $\checkmark$ 。

### • 选择题

- 5) 假设输入的字符串的所有字母都为小写字母, 则改变的字母个数为字符串长度的

约数个数, 若输入字符串长度为 18, 因为 18 的因数有: 1, 2, 3, 6, 9, 18 一共 6 个数, 则最多 6 个位置被修改, 故选 B。

6) 输入的字符串中, 被修改的字符数最多为  $n$  的因数个数, 依次观察 4 个选项, 其中  $36 = 2^2 \times 3^2$ , 计有  $(2+1) \times (2+1) = 9$  个因数,  $100000 = 2^5 \times 5^5$ , 共有  $(5+1) \times (5+1) = 36$  个因数, 1 只有一个因子,  $128 = 2^7$ , 共有 8 个因数, 故选择 B。

## 1 问题描述

```

1  #include <cstdio>
2  using namespace std;
3  int n, m;
4  int a[100], b[100];
5
6  int main() {
7      scanf("%d%d", &n, &m);
8      for (int i = 1; i <= n; ++i)
9          a[i] = b[i] = 0;
10     for (int i = 1; i <= m; ++i) {
11         int x, y;
12         scanf("%d%d", &x, &y);
13         if (a[x] < y && b[y] < x) {
14             if (a[x] > 0)
15                 b[a[x]] = 0;
16             if (b[y] > 0)
17                 a[b[y]] = 0;
18             a[x] = y;
19             b[y] = x;
20         }
21     }
22     int ans = 0;
23     for (int i = 1; i <= n; ++i) {
24         if (a[i] == 0)
25             ++ans;
26         if (b[i] == 0)
27             ++ans;
28     }
29     printf("%d\n", ans);
30     return 0;
31 }
```

假设输入的  $n$  和  $m$  都是正整数,  $x$  和  $y$  都是在  $[1, n]$  的范围内的整数, 完成下面的判断

题和单选题:

### • 判断题

- 1) 当  $m > 0$  时, 输出的值一定小于  $2n$ 。( )
- 2) 执行完第 27 行的“++ans”时, ans 一定是偶数。( )
- 3)  $a[i]$  和  $b[i]$  不可能同时大于 0。( )
- 4) 若程序执行到第 13 行时,  $x$  总是小于  $y$ , 那么第 15 行不会被执行。( )

### • 选择题

- 5) 若  $m$  个  $x$  两两不同, 且  $m$  个  $y$  两两不同, 则输出的值为( )  
A.  $2n-2m$     B.  $2n+2$     C.  $2n-2$     D.  $2n$
- 6) 若  $m$  个  $x$  两两不同, 且  $m$  个  $y$  都相等, 则输出的值为( )  
A.  $2n-2$     B.  $2n$     C.  $2m$     D.  $2n-2m$

## 2 考察的知识点

程序的模拟, 特殊值的构造。

## 3 解题思路

这道题是这样的一个模型, 有  $2n$  个点,  $X_1, X_2, \dots, X_n$  和  $Y_1, Y_2, \dots, Y_n$ 。加入  $m$  条边, 每条边只能在  $X$  和  $Y$  之间相连, 且每个点只连一条边, 若  $X_i$  与  $Y_j$  和  $Y_k$  ( $j < k$ ) 之间都有边相连, 则只保留  $X_i$  与  $Y_k$  这条边。答案为寻找没有连边的点的个数。

阅读程序, 该程序对于初始为 0 的数组, 读入  $m$  组  $x, y$ , 若  $a[x]$  被操作过则将  $b[a[x]]$  赋为 0, 若  $b[y]$  被操作过则将  $a[b[y]]$  赋为 0, 再将  $a[x]$  赋为  $y$ ,  $b[y]$  赋为  $x$ , 统计两数组中 0 的个数。

### • 判断题

- 1) 当  $m > 0$  时, 则  $m$  至少为 1, 至少有两个位置不为 0, 则  $ans < 2n$ , 可以知道  $a$  数组和  $b$  数组中一定会有值被修改, 故答案为  $\checkmark$ 。
- 2) 可以举出反例, 当  $n=2, m=2, x_1=1, y_1=1, x_2=1, y_2=2$  时, 最后的结果数组为  $a[] = \{2, 0\}, b[] = \{0, 1\}$ , 当  $i=1$  时运行到 27 行的“++ans”时,  $ans=1$ , 被修改的位置的个数为奇数, 故答案为  $\times$ 。

3) 当  $n=1, m=1, x_1=1, y_1=1$  时, 则有  $a[1]=b[1]=1 > 0$ , 故答案为  $\times$ 。

4) 当输入的数据为

3 2

1 2

1 3

程序的第 15 行被执行, 故答案为  $\times$ 。

### • 选择题

5) 当  $m$  个  $x$  两两不同, 且  $m$  个  $y$  两两不同, 则对于每次前修改  $a[x]=b[y]=0$ , 修改后  $a[x]=y, b[y]=x$ , 一次修改 2 个位置,  $m$  次后值为 0 的位置剩余  $(2n-2m)$  个。故选择 A。

6) 当  $m$  个  $x$  两两不同, 且  $m$  个  $y$  都相等, 则每次修改的  $b$  数组的位置都相同, 只有当新的  $x$  满足  $x > b[y]$  时, 可能进行修改, 修改时  $a[b[y]]=0$ , 则前一次对于  $a$  数组的修改效果被消除, 变为  $a[x]=y$ , 即在这种情况下, 只有  $x$  最大的那次修改能对  $a$  数组产生影响, 则  $a, b$  数组共有两个位置受到影响, 故答案为  $ans=2n-2$ , 选择 A。

## 1 问题描述

```

1  #include <iostream>
2  using namespace std;
3  const int maxn = 10000;
4  int n;
5  int a[maxn];
6  int b[maxn];
7  int f(int l, int r, int depth) {
8      if (l > r)
9          return 0;
10     int min = maxn, mink;
11     for (int i = l; i <= r; ++i) {
12         if (min > a[i]) {
13             min = a[i];
14             mink = i;
15         }
16     }
17     int lres = f(l, mink - 1, depth + 1);
18     int rres = f(mink + 1, r, depth + 1);
19     return lres + rres + depth * b[mink];
20 }
21 int main() {

```

```

22  cin >> n;
23  for (int i = 0; i < n; ++ i)
24      cin >> a[i];
25  for (int i = 0; i < n; ++ i)
26      cin >> b[i];
27  cout << f(0, n - 1, 1) << endl;
28  return 0;
29 }

```

### • 判断题

1) 如果 a 数组有重复的数字, 则程序运行时会发生错误。( )

2) 如果 b 数组全为 0, 则输出为 0。( )

### • 选择题

3) 当  $n=100$  时, 最坏情况下, 与第 12 行的比较运算执行的次数最接近的是( )

A. 5000    B. 6000    C. 6    D. 100

4) 当  $n=100$  时, 最好情况下, 与第 12 行的比较运算执行的次数最接近的是( )

A. 100    B. 6    C. 5000    D. 600

5) 当  $n=10$  时, 若 b 数组满足, 对任意  $0 \leq i < n$ , 都有  $b[i]=i+1$ , 那么输出最大为( )

A. 386    B. 383    C. 384    D. 385

6) (4 分) 当  $n=100$  时, 若 b 数组满足, 对任意  $0 \leq i < n$ , 都有  $b[i]=1$ , 那么输出最小为( )

A. 582    B. 580    C. 579    D. 581

## 2 考察的知识点

对分治算法、递归程序的理解, 二叉树, 程序复杂度计算。

## 3 解题思路

F 函数本质是构筑一棵二叉树, 序列中最小的数做二叉树的根, 左边的子序列做左子树, 右边的树做右子树, 然后反复做此操作, 直到构筑出整棵树。程序对 a 数组进行分治, 每次执行时找到当前区间的最小值的下标, 将这个下标为界分成左右两边再分别计算值, 最后加上递归层数乘上

$b[mink]$  进行回溯, 每个子树以该区间的  
最小值点为根, 答案就为  $\sum dep[i]*b[i]$   
( $dep[i]$  为  $i$  点在树上的深度)。

### • 判断题

1) 观察程序可知, 若 a 中有重复数字, 每次都会取下标较靠前的那个最小值, 且该程序中存在判断超界的 “if (l>r) return 0;”, 所以不会运行时错误, 故答案为  $\times$ 。

2) 由分析可知, 答案为  $dep[i]*b[i]$ 。若  $b[i]$  全为 0, 则对于函数 f 的每一个返回值一定都为 0, 故答案为  $\checkmark$ 。

### • 选择题

3) 当  $n=100$  时, 最坏情况下该数组 a 为单调递增, 每次都必须从当前区间的第一个位置比较到最后一个位置, 每次比较都要更新 min 和 mink 的值, 则需要判断  $100+99+98+\dots+2+1=5050$ , 最接近 5000, 故选择 A。

4) 当  $n=100$  时, 最好情况下, 该数组每个区间的最小值都位于区间的中点, 这样层数最小, 每次都可以将 a 数组平均分为两份。此时构建出的二叉树高度为  $\log_2(n)$ , 总复杂度为  $n*\log_2(n)$ , 当  $n=100$  时, 比较运算执行的次数为  $100*\log_2(100) \approx 600$  次, 故选择 D。

5) 当  $n=10$ ,  $0 \leq i < n$ ,  $b[i]=i+1$  时, 要使得答案最大, 则  $b[i]$  越大对应深度越大越好。构造 a[i] 为一个递增数组, 则  $0 \leq i < n$ , 时, i 的深度由 1 到 n 依次递增, 则  $1*b[0]+2*b[1]+\dots+9*b[8]+10*b[9]=1*1+2*2+3*3+\dots+10*10=385$ , 故选 D。

7) 当  $n=100$ ,  $0 \leq i < n$ ,  $b[i]=1$  时, 答案最少的情况下递归树为一颗完全二叉树, 设根节点 (原数组) 为第 0 层, 则第 1 层有 1 个划分点, 第 2 层有 2 个划分点, 第 3 层有 4 个划分点, 第 n 层有  $2^{n-1}$  个划分点, 最后一层即第 7 层剩下  $100-(2^7-1)=37$  个节点, 通过模拟运算, 可以计算出函数最终返回值为  $1*1+2*2+4*3+\dots+32*6+37*7=580$ , 故选择 B。