

# 【华为OD机考 统一考试机试C卷】字符串变换最小字符串（C++ Java JavaScript Python C语言）

## 华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷

目前在考C卷，经过两个月的收集整理，**C卷真题已基本整理完毕**  
抽到原题的概率为2/3到3/3，**也就是最少抽到两道原题。请注意：大家刷完C卷真题，最好要把B卷的真题刷一下，因为C卷的部分真题来自B卷。**  
另外订阅专栏还可以联系笔者开通在线 **OJ** 进行刷题，提高刷题效率。  
**真题目录：**华为OD机考机试 真题目录（C卷 + D卷 + B卷 + A卷） + 考点说明  
**专栏：**2023华为OD机试( B卷+C卷+D卷) (C++JavaJSPy)  
**华为OD面试真题精选：**华为OD面试真题精选  
**在线OJ：**点击立即刷题，模拟真实机考环境

## 题目描述

给定一个字符串s，最多只能进行一次变换，返回变换后能得到的最小字符串（按照字典序进行比较）。

变换规则：交换字符串中任意两个不同位置的字符。

## 输入描述

一串小写字母组成的字符串s

## 输出描述

按照要求进行变换得到的最小字符串。

## 用例

输入	abcdef
输出	abcdef

说明	abcdef已经是最小字符串，不需要交换。
输入	bcdefa
输出	acdefb
说明	a和b进行位置交换，可以得到最小字符串。

备注

s是都是小写字符组成

1<=s.length<=1000

C++

```
1  #include <iostream>
2  #include <algorithm>
3  #include <string>
4
5  int main() {
6      std::string s;
7      std::cin >> s;
8
9      // 对字符串进行排序
10     std::string sortedArr = s;
11     std::sort(sortedArr.begin(), sortedArr.end());
12
13     // 如果排序后的字符串与原字符串相同，则说明已经是最小字符串，直接输出
14     if (sortedArr == s) {
15         std::cout << s << std::endl;
16         return 0;
17     }
18
19     // 遍历原字符串
20     std::string sb = s;
21     for (int i = 0; i < s.length(); i++) {
22         // 如果当前字符与排序后的字符不相同，则进行交换
23         if (s[i] != sortedArr[i]) {
24             swap(s[i], sortedArr[i]);
25         }
26     }
```

```

24         char tmp = sb[i];
25         int swapIndex = -1;
26         // 找到排序后的字符在原字符串中的位置
27         for (int j = i + 1; j < s.length(); j++) {
28             if (sb[j] == sortedArr[i]) {
29                 swapIndex = j;
30             }
31         }
32         // 将原字符与排序后的字符交换
33         sb[i] = sortedArr[i];
34         sb[swapIndex] = tmp;
35         break;
36     }
37 }
38
39 // 输出最小字符串
40 std::cout << sb << std::endl;
41
42 return 0;
43 }
44
45

```

## JavaScript

```

1  const readline = require('readline');
2
3  const rl = readline.createInterface({
4      input: process.stdin,
5      output: process.stdout
6  });
7
8  rl.on('line', (s) => {
9      // 对字符串进行排序
10     const sortedArr = s.split('').sort();
11
12     // 如果排序后的字符串与原字符串相同，则说明已经是最小字符串，直接输出
13     if (sortedArr.join('') === s) {
14         console.log(s);
15         rl.close();
16     }
17 }

```

```

16     return;
17 }
18
19 // 遍历原字符串
20 let sb = s.split('');
21 for (let i = 0; i < s.length; i++) {
22     // 如果当前字符与排序后的字符不相同, 则进行交换
23     if (s.charAt(i) !== sortedArr[i]) {
24         const tmp = sb[i];
25         let swapIndex = -1;
26         // 找到排序后的字符在原字符串中的位置
27         for (let j = i + 1; j < s.length; j++) {
28             if (sb[j] === sortedArr[i]) {
29                 swapIndex = j;
30             }
31         }
32         // 将原字符与排序后的字符交换
33         sb[i] = sortedArr[i];
34         sb[swapIndex] = tmp;
35         break;
36     }
37 }
38
39 // 输出最小字符串
40 console.log(sb.join(''));
41 r1.close();
42 });

```

## Java

```

1 import java.util.Arrays;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         String s = sc.nextLine();
8
9         // 对字符串进行排序
10        char[] sortedArr = s.toCharArray();
11

```

```

11 Arrays.sort(sortedArr);
12
13 // 如果排序后的字符串与原字符串相同, 则说明已经是最小字符串, 直接输出
14 if (new String(sortedArr).equals(s)) {
15     System.out.println(s);
16     return;
17 }
18
19 // 遍历原字符串
20 StringBuilder sb = new StringBuilder(s);
21 for (int i = 0; i < s.length(); i++) {
22     // 如果当前字符与排序后的字符不相同, 则进行交换
23     if (s.charAt(i) != sortedArr[i]) {
24         char tmp = sb.charAt(i);
25         int swapIndex = -1;
26         // 找到排序后的字符在原字符串中的位置
27         for (int j = i + 1; j < s.length(); j++) {
28             if (sb.charAt(j) == sortedArr[i]) {
29                 swapIndex = j;
30             }
31         }
32         // 将原字符与排序后的字符交换
33         sb.setCharAt(i, sortedArr[i]);
34         sb.setCharAt(swapIndex, tmp);
35         break;
36     }
37 }
38
39 // 输出最小字符串
40 System.out.println(sb.toString());
41 }
42 }
43

```

## python

```

1 s = input()
2
3 # 对字符串进行排序
4 sortedArr = sorted(s)
5

```

```

5
6 # 如果排序后的字符串与原字符串相同, 则说明已经是最小字符串, 直接输出
7 if ''.join(sortedArr) == s:
8     print(s)
9     exit()
10
11 # 遍历原字符串
12 sb = list(s)
13 for i in range(len(s)):
14     # 如果当前字符与排序后的字符不相同, 则进行交换
15     if s[i] != sortedArr[i]:
16         tmp = sb[i]
17         swapIndex = -1
18         # 找到排序后的字符在原字符串中的位置
19         for j in range(i + 1, len(s)):
20             if sb[j] == sortedArr[i]:
21                 swapIndex = j
22         # 将原字符与排序后的字符交换
23         sb[i] = sortedArr[i]
24         sb[swapIndex] = tmp
25         break
26
27 # 输出最小字符串
28 print(''.join(sb))
29
30

```

## C语言

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4
5  #define MAX_STR_LEN 1000
6
7  // 定义比较函数, 用于 qsort 函数
8  int cmp(const void *a, const void *b) {
9      return *(char *)a - *(char *)b;
10 }
11
12

```

```

12 int main() {
13     char s[MAX_STR_LEN], sortedArr[MAX_STR_LEN];
14     scanf("%s", s);
15
16     // 对字符串进行排序
17     strcpy(sortedArr, s);
18     qsort(sortedArr, strlen(sortedArr), sizeof(char), cmp);
19
20     // 如果排序后的字符串与原字符串相同，则说明已经是最小字符串，直接输出
21     if (strcmp(sortedArr, s) == 0) {
22         printf("%s\n", s);
23         return 0;
24     }
25
26     // 遍历原字符串
27     for (int i = 0; i < strlen(s); i++) {
28         // 如果当前字符与排序后的字符不相同，则进行交换
29         if (s[i] != sortedArr[i]) {
30             char tmp = s[i];
31             int swapIndex = -1;
32             // 找到排序后的字符在原字符串中的位置
33             for (int j = i + 1; j < strlen(s); j++) {
34                 if (s[j] == sortedArr[i]) {
35                     swapIndex = j;
36                 }
37             }
38             // 将原字符与排序后的字符交换
39             s[i] = sortedArr[i];
40             s[swapIndex] = tmp;
41             break;
42         }
43     }
44
45     // 输出最小字符串
46     printf("%s\n", s);
47
48     return 0;
49 }

```

## 完整用例

### 用例1

abcdef

### 用例2

bcdefa

### 用例3

aaa

### 用例4

cba

### 用例5

dcba

### 用例6

abccba

### 用例7

abacaba

### 用例8

abcde

### 用例9

aaaaaa

### 用例10

dcbaabc



## 文章目录

华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷

题目描述

输入描述

输出描述

用例

备注

c++

javaScript

Java

python

C语言

完整用例

用例1

用例2

用例3

用例4

用例5

用例6

用例7

用例8

用例9

用例10

# 机考真题 华为OD



CSDN @算法大师