

2012腾讯笔试的一道算法题

题目以及要求：把一个字符串的大写字母放到字符串的后面，各个字符的相对位置不变，不能申请额外的空间。

我的实现类似冒泡排序。

```
#include <stdio.h>
#include <string.h>

// Author: 397090770
// E-mail:wyphao.2007@163.com
// Blog:
// Date: 2012/09/29

//题目以及要求：把一个字符串的大写字母放到字符串的后面，
//各个字符的相对位置不变，不能申请额外的空间。

//判断是不是大写字母
int isUpperAlpha(char c){
    if(c >= 'A' && c <= 'Z'){
        return 1;
    }
    return 0;
}

//交换两个字母
void swap(char *a, char *b){
    char temp = *a;
    *a = *b;
    *b = temp;
}

char * mySort(char *arr, int len){
    if(arr == NULL || len <= 0){
        return NULL;
    }

    int i = 0, j = 0, k = 0;
    for(i = 0; i < len; i++){
        for(j = len - 1 - i; j >= 0; j--){
            if(isUpperAlpha(arr[j])){
                for(k = j; k < len - i - 1; k++){
                    swap(&arr[k], &arr[k + 1]);
                }
            }
        }
    }
}
```

```
break;
}
//遍历完了字符数组，但是没发现大写字母，所以没必要再遍历下去
if(j == 0 && !isUpperAlpha(arr[j])){
    //goto over;
    return arr;
}
}
}

//over:
return arr;
}

int main(){
    char arr[] = "aaaaaaaaaaaaaaaaaaaaaAbcAdeBbDc";
    printf("%s\n", mySort(arr, strlen(arr)));
    return 0;
}
```

运行结果：aaaaaaaaaaaaaaaaaaaaaabcdebcAABD

后来想了一会，优化了一下代码（好像没优化，和上面的差不多），如下：

```
#include <stdio.h>
#include <string.h>

// Author: 397090770
// E-mail:wyphao.2007@163.com
// Blog:
// Date: 2012/09/29

//题目以及要求：把一个字符串的大写字母放到字符串的后面，
//各个字符的相对位置不变，不能申请额外的空间。

//判断是不是大写字母
int isUpperAlpha(char c){
    if(c >= 'A' && c <= 'Z'){
        return 1;
    }
    return 0;
}

//交换两个字母
```

```
void swap(char *a, char *b){
    char temp = *a;
    *a = *b;
    *b = temp;
}

char * mySort(char *arr, int len){
    if(arr == NULL || len <= 0){
        return NULL;
    }

    int i = 0, j = 0, k = 0;
    //for(i = 0; i < len; i++){ for(j = len - 1 - i; j >= 0; j--){
    if(isUpperAlpha(arr[j])){
        for(k = j; k < len - i - 1; k++){
            swap(&arr[k], &arr[k + 1]);
        }
        i++;
        j = len - 1 - i;
        //break;
    }
    //遍历完了字符数组，但是没发现大写字母，所以没必要再遍历下去
    if(j == 0 && !isUpperAlpha(arr[j])){
        //goto over;
        return arr;
    }
}
//over:
return arr;
}

int main(){
    char arr[] = "GaaaaBGaaaaaaaaaaaaaaaaAbcAdeBbDc";
    printf("%s\n", mySort(arr, strlen(arr)));
    return 0;
}
```

在第二个for循环里面，每一次都从尾部开始，让费了时间，所以我设置了一个变量，来保存找到大写字母的位置，这样在下次遍历的时候只需要从这里开始向前遍历：

```
int tempIndex = len - 1;
```

```
for(i = 0; i < len; i++){ for(j = tempIndex; j >= 0; j--){
  if(isUpperAlpha(arr[j])){
    tempIndex = j - 1;
    for(k = j; k < len - i - 1; k++){
      swap(&arr[k], &arr[k + 1]);
    }
    break;
  }
  //遍历完了字符数组，但是没发现大写字母，所以没必要再遍历下去
  if(j == 0 && !isUpperAlpha(arr[j])){
    return arr;
  }
}
}
```

(转载请注明：</archives/132>，请不要用于商业目的。)

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接：[【】（）](#)