

亲密字符串

给定两个由小写字母构成的字符串 A 和 B，只要我们可以通过交换 A 中的两个字母得到与 B 相等的结果，就返回 true；否则返回 false。

交换字母的定义是取两个下标 i 和 j（下标从 0 开始），只要 $i \neq j$ 就交换 A[i] 和 A[j] 处的字符。例如，在 "abcd" 中交换下标 0 和下标 2 的元素可以生成 "cbad"。

示例 1:

输入: A = "ab", B = "ba"

输出: true

解释: 你可以交换 A[0] = 'a' 和 A[1] = 'b' 生成 "ba", 此时 A 和 B 相等。

示例 2:

输入: A = "ab", B = "ab"

输出: false

解释: 你只能交换 A[0] = 'a' 和 A[1] = 'b' 生成 "ba", 此时 A 和 B 不相等。

示例 3:

输入: A = "aa", B = "aa"

输出: true

解释: 你可以交换 A[0] = 'a' 和 A[1] = 'a' 生成 "aa", 此时 A 和 B 相等。

示例 4:

输入: A = "aaaaaaabc", B = "aaaaaaacb"

输出: true

示例 5:

输入: A = "", B = "aa"

输出: false

解题思路:

1. 首先我们知道两个字符串要么是亲密字符串,要么不是亲密字符串
2. 对于亲密字符串来说,有两种情况,一种是两个字符串完全相同,此时如果字符串中有两个相同的字母那么进行交换以后字符串不变,成为亲密字符串的条件成立. 另一种是两个字符串不同,此时字符串应该存在两处不同(一处或者三处都无法成为亲密字符串),进行一次交换便可以变成另一个字符串
3. 对于非亲密字符串的话 一种就是如果 A B 长度不同那么一定不是亲密字符串. 一种是两个字符串有一种或者二种以上的不同的地方(或者两种地方不同但是不满足 $A[i] == B[j]$ $B[i] == A[j]$).最后一种是两个字符串完全相同,但是没有有一个字符出现两次以上

- 1.用 count 记录不同的数组下标;
- 2.count > 2 或 count = 1 则返回 false;
- 3.count = 2 时判断两个互换是否相等;
- 4.count = 0 判断 A 中是否重复元素出现,有返回 true 否则 false;

```
bool buddyStrings(char * A, char * B){
    int lena=strlen(A);
    int lenb=strlen(B);
    int cout=0;
    int tmp[3]={0};
    if(lena==0&&lenb==0)//有一方为空肯定无法变换得到相同
    {
        return false;
    }
    if(lena!=lenb)//不等长肯定无法变换得到相同
    {
        return false;
    }
    //此时 A,B 等长情况下
    for(int i=0;i<lena;i++)
    {
        if(A[i]-B[i]!=0)
        {
            tmp[cout++]=i;//记录不相同位置,同时计数不相同次数
        }
    }
}
```

```
    //如果不相同位置超过两处 则直接返回 false
    if(cout>2)
    {
        return false;
    }
}
if(cout==1)//只有 1 处不相同则直接返回 false
{
    return false;
}
if(cout==2&&A[tmp[0]]==B[tmp[1]]&&A[tmp[1]]==B[tmp[0]])//两处不相同，且
两处互换相等
{
    return true;
}
if(cout==0)//二者对应位置完全相同 需要判断是否其中有重复项 有则 true 无则
false
{
    for(int i=0;i<lena;i++)
    {
        for(int j=i+1;j<lena;j++)
        {
            if(A[i]==A[j])
                return true;
        }
    }
}
return false;
}
```