

本章字符串和链表的习题

1、第一个只出现一次的字符

在一个字符串中找到第一个只出现一次的字符。如输入abaccdeff，则输出b。

2、对称子字符串的最大长度

输入一个字符串，输出该字符串中对称的子字符串的最大长度。比如输入字符串“google”，由于该字符串里最长的对称子字符串是“goog”，因此输出4。

提示：可能很多人都写过判断一个字符串是不是对称的函数，这个题目可以看成是该函数的加强版。

3、编程判断两个链表是否相交

给出两个单向链表的头指针，比如h1，h2，判断这两个链表是否相交。为了简化问题，我们假设两个链表均不带环。

问题扩展：

- 如果链表可能有环列？
- 如果需要求出两个链表相交的第一个节点列？

4、逆序输出链表

输入一个链表的头结点，从尾到头反过来输出每个结点的值。

5、在O(1)时间内删除单链表结点

给定单链表的一个结点的指针，同时该结点不是尾结点，此外没有指向其它任何结点的指针，请在O(1)时间内删除该结点。

6、找出链表的第一个公共结点

两个单向链表，找出它们的第一个公共结点。

7、在字符串中删除特定的字符

输入两个字符串，从第一个字符串中删除第二个字符串中所有的字符。

例如，输入“They are students.”和“aeiou”，则删除之后的第一个字符串变成“Thy r stdnts.”。

8、字符串的匹配

在一篇英文文章中查找指定的人名，人名使用二十六个英文字母（可以是大写或小写）、空格以及两个通配符组成（、？），通配符“*”表示零个或多个任意字母，通配符“?”表示一个任意字母。如：“J* Smi??”可以匹配“John Smith”。

9、字符个数的统计

char *str = "AbcABca"; 写出一个函数，查找出每个字符的个数，区分大小写，要求时间复杂度是n（提示用ASCII码）

10、最小子串

给一篇文章，里面是由一个个单词组成，单词中间空格隔开，再给一个字符串指针数组，比如 `char *str[] = {"hello", "world", "good"};`

求文章中包含这个字符串指针数组的最小子串。注意，只要包含即可，没有顺序要求。

提示：文章也可以理解为一个大的字符串数组，单词之前只有空格，没有标点符号。

11、字符串的集合

给定一个字符串的集合，格式如：{aaa bbb ccc}，{bbb ddd}，{eee fff}，{ggg}，{ddd hhh}要求将其中交集不为空的集合合并，要求合并完成后的集合之间无交集，例如上例应输出{aaa bbb ccc ddd hhh}，{eee fff}，{ggg}。

提示：并查集。

12、五笔编码

五笔的编码范围是a ~ y的25个字母，从1位到4位的编码，如果我们把五笔的编码按字典序排序，形成一个数组如下：a, aa, aaa, aaaa, aaab, aaac,, b, ba, baa, baaa, baab, baac, yyyw, yyyx, yyyy 其中a的Index为0，aa的Index为1，aaa的Index为2，以此类推。

- 编写一个函数，输入是任意一个编码，比如baca，输出这个编码对应的Index；
- 编写一个函数，输入是任意一个Index，比如12345，输出这个Index对应的编码。

13、最长重复子串

一个长度为10000的字符串，写一个算法，找出最长的重复子串，如abczzacbca,结果是bc。

提示：此题是后缀树/数组的典型应用，即是求后缀数组的height[]的最大值。

14、字符串的压缩

一个字符串，压缩其中的连续空格为1个后，对其中的每个字串逆序打印出来。比如"abc efg hij"打印为"cba gfe jih"。

15、最大重复出现子串

输入一个字符串，如何求最大重复出现的字符串呢？比如输入ttabcftgabcd,输出结果为abc, canffcancd,输出结果为can。

给定一个字符串，求出其最长的重复子串。

分析：使用后缀数组，对一个字符串生成相应的后缀数组后，然后再排序，排完序依次检测相邻的两个字符串的开头公共部分。这样的时间复杂度为：

- 生成后缀数组 $O(N)$
- 排序 $O(N \log N)$ 最后面的 N 是因为字符串比较也是 $O(N)$
- 依次检测相邻的两个字符串 $O(N * N)$

故最终总的时间复杂度是 $O(N^2 \log N)$

16、字符串的删除

删除模式串中出现的字符，如“welcome to asted”,模式串为“aeiou”那么得到的字符串为“wlcm t std”，要求性能最优。

17、字符串的移动

字符串为*号和26个字母的任意组合，把*号都移动到最左侧，把字母移到最右侧并保持相对顺序不变，要求时间和空间复杂度最小。

18、字符串的包含

输入：

L:“hello”“july”

S:“hellomehellojuly”

输出：S中包含的L一个单词，要求这个单词只出现一次，如果有多个出现一次的，输出第一个这样的单词。

19、倒数第n个元素

链表倒数第n个元素。

提示：设置一前一后两个指针，一个指针步长为1，另一个指针步长为n，当一个指针走到链表尾端时，另一指针指向的元素即为链表倒数第n个元素。

20、回文字符串

将一个很长的字符串，分割成一段一段的子字符串，子字符串都是回文字符串。有回文字符串就输出最长的，没有回文就输出一个一个的字符。

例如：

habbafgh

输出h,abba,f,g,h。

提示：一般的人会想到用后缀数组来解决这个问题。

21、最长连续字符

用递归算法写一个函数，求字符串最长连续字符的长度，比如aaaabbcc的长度为4，aabb的长度为2，ab的长度为1。

22、字符串反转

实现字符串反转函数。

22、字符串压缩

通过键盘输入一串小写字母(a~z)组成的字符串。请编写一个字符串压缩程序，将字符串中连续出席的重复字母进行压缩，并输出压缩后的字符串。压缩规则：

- 仅压缩连续重复出现的字符。比如字符串“abcbcb”由于无连续重复字符，压缩后的字符串还是“abcbcb”。
- 压缩字段的格式为“字符重复的次数+字符”。例如：字符串“xxxxyyyyyyz”压缩后就成为“3x6yz”。

要求实现函数： `void stringZip(const char *pInputStr, long lInputLen, char *pOutputStr);`

- 输入 `pInputStr`： 输入字符串 `lInputLen`： 输入字符串长度
- 输出 `pOutputStr`： 输出字符串，空间已经开辟好，与输入字符串等长；

注意：只需要完成该函数功能算法，中间不需要有任何IO的输入输出

示例

- 输入：“cccddecc” 输出：“3c2de2c”
- 输入：“adef” 输出：“adef”
- 输入：“ppppppppp” 输出：“8p”

23、集合的差集

已知集合A和B的元素分别用不含头结点的单链表存储，请求集合A与B的差集，并将结果保存在集合A的单链表中。例如，若集合A={5,10,20,15,25,30}，集合B={5,15,35,25}，完成计算后A={10,20,30}。

24、最长公共子串

给定字符串A和B，输出A和B中的第一个最长公共子串，比如A="wepiabc B="pabcni"，则输出"abc"。

25、均分01

给定一个字符串，长度不超过100，其中只包含字符0和1,并且字符0和1出现得次数都是偶数。你可以把字符串任意切分，把切分后得字符串任意分给两个人，让两个人得到的0的总个数相等，得到的1的总个数也相等。

例如，输入串是010111,我们可以把串切位01, 011,和1，把第1段和第3段放在一起分给一个人，第二段分给另外一个人，这样每个人都得到了1个0和两个1。我们要做的是让切分的次数尽可能少。

考虑到最差情况，则是把字符串切分(n - 1)次形成n个长度为1的串。

26、合法字符串

用n个不同的字符（编号1 - n），组成一个字符串，有如下2点要求：

- 1、对于编号为i的字符，如果 $2 * i > n$ ，则该字符可以作为最后一个字符，但如果该字符不是作为最后一个字符的话，则该字符后面可以接任意字符；
- 2、对于编号为i的字符，如果 $2 * i \leq n$ ，则该字符不可以作为最后一个字符，且该字符后面所紧接着的下一个字符的编号一定要 $\geq 2 * i$ 。

问有多少长度为M且符合条件的字符串。

例如：N = 2，M = 3。则abb, bab, bbb是符合条件的字符串，剩下的均为不符合条件的字符串。

假定n和m皆满足： $2 \leq n, m \leq 1000000000$ 。

27、最短摘要生成

你我在百度或谷歌搜索框中敲入本博客名称的前4个字“结构之法”，便能在第一个选项看到本博客的链接，如下图2所示：

在上面所示的图2中，搜索结果“结构之法算法之道-博客频道-CSDN.NET”下有一段说明性的文字：“程序员面试、算法研究、编程艺术、红黑树4大经典原创系列集锦与总结 作者：July--结构之法算法...”，我们把这段文字称为那个搜索结果的摘要，亦即最短摘要。我们的问题是，请问，这个最短摘要是怎么生成的呢？

28、实现memcpy函数

已知memcpy的函数为：`void* memcpy(void* dest , const void* src , size_t count)`其中dest是目的指针，src是源指针。不调用c++/c的memcpy库函数，请编写memcpy。

分析：参考代码如下：

```
void* memcpy(void *dst, const void *src, size_t count)
{
    //安全检查
    assert( (dst != NULL) && (src != NULL) );

    unsigned char *pdst = (unsigned char *)dst;
    const unsigned char *psrc = (const unsigned char *)src;

    //防止内存重复
    assert(!(psrc<=pdst && pdst<psrc+count));
    assert(!(pdst<=psrc && psrc<pdst+count));

    while(count-->0)
    {
        *pdst = *psrc;
        pdst++;
        psrc++;
    }
    return dst;
}
```

29、实现memmove函数

分析：memmove函数是<string.h>的标准函数，其作用是把从source开始的num个字符拷贝到destination。最简单的方法是直接复制，但是由于它们可能存在内存的重叠区，因此可能覆盖了原有数据。

比如当`source+count>=dest&&source<dest`时，dest可能覆盖了原有source的数据。解决办法是从后往前拷贝，对于其它情况，则从前往后拷贝。

参考代码如下：

```
//void * memmove ( void * destination, const void * source, size_t num );
void* memmove(void* dest, void* source, size_t count)
{
    void* ret = dest;

    if (dest <= source || dest >= (source + count))
    {

```

```
        //正向拷贝
        //copy from lower addresses to higher addresses
        while (count --)
            *dest++ = *source++;
    }
    else
    {
        //反向拷贝
        //copy from higher addresses to lower addresses
        dest += count - 1;
        source += count - 1;

        while (count--)
            *dest-- = *source--;
    }
    return ret;
}
```