

数据结构 A

作业 6 参考答案

作业情况

教材：数据结构教程（C++ 语言描述）李春葆等

题目范围：串

邮箱：wjyyy1@126.com

授课教师：彭蓉 教授

助教：王骏骁

习题 1 单选题

以下关于串的叙述中正确的是（ ）

- A. 串是一种特殊的线性表
- B. 串中元素只能是字母
- C. 空串就是空白串
- D. 串的长度必须大于零

答案：A

解析：A 选项，串实际上可以认为是“字符数组”，也就是一种线性表，它比线性表的要求更严格。

B 选项，字符包含“字母、数字和其他字符”，这里的其他字符可以是标点符号等。同时广义的字符也可以是汉字。

C 选项，空串是长度为 0 的串，而空白串可以指的是只包含空格的串。

D 选项，串的长度可以为 0，也就是空串。

习题 2 单选题

两个字符串相等的条件是（ ）

- A. 串的长度相等
- B. 含有相同的字符集
- C. 串的长度相等并且对应的字符相同
- D. 串的长度和对应字符都一致并且存储结构相同

答案：C

解析：串只关注内容，不关注存储结构。只需要保证逐个对应字符都相等即可。

习题 3 单选题

若串 $S = \text{"software"}$ ，其不同子串的个数 ()

- A. 37
- B. 36
- C. 8
- D. 9

答案：A

解析：参考例 4.1，子串的定义是串内部连续的一段字符，而“software”内部的字符两两不同，因此不用考虑重复的情况（注意如果有重复的话按题目要求确定是否统计）。

长度为 1 的子串有 8 个（分别为“s”、“o”、“f”、“t”、“w”、“a”、“r”、“e”）。

长度为 2 的子串有 7 个（分别为“so”、“of”、“ft”、“tw”、“wa”、“ar”、“re”）。

长度为 3 的子串有 6 个（分别为“sof”、“oft”、“ftw”、“twa”、“war”、“are”）。

.....

长度为 7 的子串有 2 个（分别为“softwar”、“oftware”）。

长度为 8 的子串有 1 个（分别为“software”）。

不要忘记还有 1 个空串。

$$8 + 7 + 6 + \cdots + 2 + 1 + 1 = 37$$

习题 4 单选题

在 KMP 模式匹配中用 $next$ 数组存放模式串的部分匹配信息，当模式串位 j 与目标串 i 比较时两字符不相等，则 i 的位移方式是 ()

- A. $i = next[j]$
- B. i 不变
- C. $i = 0$
- D. $i = i - j + 1$

答案：B

解析： $next$ 数组存放的是最长相同前后缀。也就是当前的串如果失配了，可以尝试某个和当前后缀相等的前缀，保留一部分已经匹配的内容。

因此当模式串位 j 出现失配时， j 自行变为 $next[j]$ ，仍然与 i 进行比较。注意审题 !!! 减少被题目误导的机会。

习题 5 两串的最长相同前后缀**【问题描述】**

给定两个字符串 s_1 和 s_2 ，求最长的 s_1 前缀 ss 使得 ss 为 s_2 的最长后缀，输出该字符串和其长度。

【输入形式】

输入的两个测试用例由两行构成，第一行为 s_1 ，第二行为 s_2 。假设所有输入的数据均为小写字母。

【输出形式】

每个测试用例的输出由单行组成，其中包含最长的字符串，该字符串是 s_1 的前缀和 s_2 的后缀，后面跟着该前缀的长度。如果最长的此类字符串是空字符串，则输出应为 0。 s_1 和 s_2 的长度最多为 50000。

【样例输入】

aaariemann marjorieaaa

【样例输出】

aaa 3

【样例说明】

输入的第一行字符串 $s_1 = \text{'aaariemann'}$ ，第二行字符串 $s_2 = \text{'marjorieaaa'}$ 。 s_1 的前缀和 s_2 的后缀最长相等字符串为 aaa，因此输出 aaa 3，而不是 a 1。测试数据存放在 in.txt 文件中。

【评分标准】

该题目有 10 个测试用例，每通过一个测试用例，得 10 分。

答案：

```
1 #include<iostream>
2 #include<fstream>
3 using namespace std;
4 int nextval[100100];
5 int main()
6 {
7     ifstream in("in.txt");
8     string s1,s2;
9     in>>s1>>s2;
10    s1+=s2;//把s2连接到s1后面
11    //求新s1的next数组
12    int i=0,k=-1;
13    nextval[0]=-1;
14    while(i<s1.length())
15    {
16        if(k==-1||s1[i]==s1[k])
```

```
17     {
18         i++;
19         k++;
20         if(s1[i]==s1[k])
21             nextval[i]=k;
22         else
23             nextval[i]=nextval[k];
24     }
25     else
26         k=nextval[k];
27 }
28 int common=nextval[s1.length()-1]+1;
29 if(common>min(s1.length(),s2.length()))
30     common=min(s1.length(),s2.length());
31 cout<<s1.substr(0,common)<<" "<<common;
32 return 0;
33 }
```

解析：本题需要求出“ s_1 的前缀与 s_2 的后缀相等的最长长度”。最简单的方法是逐个比较 $s_1[0..i]$ 和 $s_2[length-i-1..length-1]$ 是否相等。但是这样的时间复杂度是 $O(n^2)$ ，正常情况下会超时（查阅大家代码后发现这样做也没有超时）。

实际上本题是需要大家使用 KMP 算法的 *nextval* 数组。首先将 s_2 连接到 s_1 后面，然后求出新串的 *next* 数组。

$nextval[i]$ 的含义就是 $\max\{k|0 < k < i, "s_0s_1 \dots s_{k-1}" = "s_{i-k}s_{i-k+1} \dots s_{i-1}"\}$ ，刚好对应了 $s_1 + s_2$ （此处 + 指连接操作）的前缀和后缀相等的情况。

最后输出长度 $nextval[s_1.length-1] + 1$ 即可。

但是要注意，拼接后的 *next* 值是有可能超出 s_1 和 s_2 的长度的，思考 $s_1 = \text{"aaaaaa"}$ 和 $s_2 = \text{"aaaaaa"}$ 拼接的情况。所以要判断使结果不超过 $\min(s_1.length, s_2.length)$ 。

习题 6 压缩字符串

【问题描述】

给定一组字符，使用原地算法将其压缩。压缩后的长度必须始终小于或等于原数组长度。数组的每个元素应该是长度为 1 的字符（不是 `int` 整数类型）。在完成原地修改输入数组后，返回数组的新长度。

【输入形式】

输入一组字符，字符间以空格隔开。

【输出形式】

输出一组字符，重复的字符用重复次数替代。每个数字在数组中都有它的位置。输出字符间以空格隔开。

【样例输入】

a a b b c c c

【样例输出】

a 2 b 2 c 3

【样例说明】

“a a”被“a 2”替代。“b b”被“b 2”替代。“c c c”被“c 3”替代。测试数据存放在 in.txt 文件中。

【评分标准】

该题目有 10 个测试用例，每通过一个测试用例，得 10 分。

答案：

```
1 #include<iostream>
2 #include<fstream>
3 #include<stack>
4 using namespace std;
5 stack<int> s;
6 int main()
7 {
8     ifstream in("in.txt");
9     char c,pre=0;
10    int cnt=0;
11    while(in>>c) // 逐字符输入，没有字符了就停止（输入流会忽略空格）
12    {
13        if(c==pre) // 如果仍然相等，就继续统计
14            cnt++;
15        else
16        {
17            if(pre!=0) // pre=0 表示是第一个字符
18            {
19                cout<<pre<<" ";
20                if(cnt>1) // cnt=1 时就不输出数量
21                {
22                    while(cnt) // 使用栈把字符数量分空格输出
23                    {
24                        s.push(cnt%10);
25                        cnt/=10;
26                    }
27                    while(!s.empty()) // 最高位会在栈顶
28                    {
29                        cout<<s.top()<<" ";
30                        s.pop();
31                    }
32                }
33            }
```

```
34         cnt=1;
35         pre=c;
36     }
37 }
38
39 cout<<pre<<" "; // 注意最后一个字符在结尾处，无法被作为pre判断
40 if(cnt>1)
41 {
42     while(cnt)
43     {
44         s.push(cnt%10);
45         cnt/=10;
46     }
47     while(!s.empty())
48     {
49         cout<<s.top()<<" ";
50         s.pop();
51     }
52 }
53 return 0;
54 }
```

解析：本题把串以字符的形式分开输入，考察对相同字符的判断。当存在相同字符时才用数字把出现的数量替代，从而满足题目中始终小于或等于原数组长度的要求。注意当数量 ≥ 10 时需要把每位数字隔开，因为每次只能方便地用取模获得最低位，所以还要用栈将这个顺序取反。

总结

题目：串

日期：2024 年 4 月 22 日（于江汉路）

批改人：王骏骁

邮箱：wjyyy1@126.com

习题 1：概念题，注意概念细节、定义区分，善于使用排除法。

习题 2：串是特殊的线性表，而对串进行比较只需要管内容是否相等即可。

习题 3：本题一方面要注意是否有相等字符（子串），如果有就需要排除。此外还有空串的情况需要考虑。和集合的概念类似，如果题目提到“真子串”也要注意不能把整个字符串算进去。

习题 3 思考题：字符串“hongkong”的不同非空真子串一共有多少个？

习题 4：注意区分模式串和匹配串的定义，在具体实现时区分函数 `GetNextval()` 和 `KMPval()`。

习题 5：本题使用枚举匹配是可以通过的，但是更建议大家向 KMP 算法的方向思考。

习题 6：注意本题的测试数据中，输入数据是由空格分开的串，此外输出的数字也是由空格分开的。大家平时写作业遇到问题时可以找助教帮大家尽快排除问题。

本次作业经查重，无代码雷同情况，感谢同学们的辛苦付出！

各位同学如有问题欢迎及时在群里提出，或者通过邮件/QQ 联系我。