

```
1  /*
2  给定一个非空字符串，其中包含字母顺序打乱的英文单词表示的数字0-9。按升序输出原始的数字。
3
4  注意：
5
6      输入只包含小写英文字母。
7      输入保证合法并可以转换为原始的数字，这意味着像 "abc" 或 "zerone" 的输入是不允许的。
8      输入字符串的长度小于 50,000。
9
10 示例 1:
11
12 输入: "owoztneoe"
13
14 输出: "012" (zeroonetwo)
15
16 示例 2:
17
18 输入: "fviefuro"
19
20 输出: "45" (fourfive)
21
22 来源：力扣（LeetCode）
23 链接：https://leetcode-cn.com/problems/reconstruct-original-digits-from-english
24 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
25 */
```

分析:

- 解题过程和解线性方程组的差不多.
- 方法一出错原因二刷再说啦.

方法一:C++\_map

```
1  class Solution
2  {
3      public:
4          string originalDigits(string s)
5          {
6
7              " f v i e f u r o "
8
9              /*
10             a b c d [e] [f] [g] [h] [i] j k l m [n] [o] p q [r] [s] [t]
11             [u] [v] [w] [x] y [z]
12
13             */
14             /*
```

```

14         zero          e:1          o:1 r:1
15             z:1
16         one          e:1          n:1 o:1
17         two          o:1          t:1
18         w:1
19         three        e:2          h:1          r:1          t:1
20         four          f:1          o:1
21         u:1
22         five          e:1 f:1          i:1
23         v:1
24         six           i:1          s:1
25         x:1
26         seven        e:2          n:1          s:1
27         v:1
28         eight        e:1          g:1 h:1          i:1          t:1
29         nine         e:1          i:1 n:2
30         */
31
32         string        charBuf("efghinorstuvwxz") ;
33         int           numCount[10] = {0,} ;
34         int           i           = 0      ;
35         int           temp        = 0      ;
36         map<char,int> mci          ;
37         string        ret_val     ;
38
39         for(i = 0 ; i < charBuf.size() ; i++)
40         {
41             mci[charBuf[i]]=0;
42         }
43
44         /*统计资源数量*/
45         for(i = 0;i < s.size();i++)
46         {
47             mci[s[i]]++;
48         }
49
50         // 1.根据'z'的数量算zero的数量,e:1 o:1 r:1
51         temp          = mci['z'] ;
52         numCount[0] = temp ;
53         mci['z']     = 0 ;
54         mci['e']      -= temp ;
55         mci['o']      -= temp ;
56         mci['r']      -= temp ;
57
58         // 2.根据'w'的数量算two的数量, o:1 t:1
59         temp          = mci['w'] ;
60         numCount[2] = temp ;
61         mci['w']     = 0 ;
62         mci['o']      -= temp ;
63         mci['t']      -= temp ;
64
65         // 3.根据'u'的数量算four的数量, f:1 o:1
66         temp          = mci['u'] ;
67         numCount[4] = temp ;
68         mci['u']     = 0 ;
69         mci['f']      -= temp ;

```

```

65     mci['o']    -=    temp        ;
66
67     // 4.根据'x'的数量算six的数量,i:1 s:1
68     temp        =    mci['x']    ;
69     numCount[6] =    temp        ;
70     mci['x']    =    0            ;
71     mci['i']    -=    temp        ;
72     mci['s']    -=    temp        ;
73
74     // 5.根据'r'的数量算three的数量,e:2 h:1 t:1
75     temp        =    mci['r']    ;
76     numCount[3] =    temp        ;
77     mci['r']    =    0            ;
78     mci['e']    -=    2*temp      ;
79     mci['h']    -=    temp        ;
80     mci['t']    -=    temp        ;
81
82     // 6.根据'o'的数量算one的数量,e:1 n:1
83     temp        =    mci['o']    ;
84     numCount[1] =    temp        ;
85     mci['o']    =    0            ;
86     mci['e']    -=    temp        ;
87     mci['n']    -=    temp        ;
88
89     // 7.根据's'的数量算seven的数量, e:2 n:1 v:1
90     temp        =    mci['s']    ;
91     numCount[7] =    temp        ;
92     mci['s']    =    0            ;
93     mci['e']    -=    2*temp      ;
94     mci['n']    -=    temp        ;
95     mci['v']    -=    temp        ;
96
97
98     // 8.根据'v'的数量算five的数量,e:1 f:1 i:1
99     temp        =    mci['v']    ;
100    numCount[5] =    temp        ;
101    mci['v']    =    0            ;
102    mci['e']    -=    temp        ;
103    mci['f']    -=    temp        ;
104    mci['i']    -=    temp        ;
105
106    // 9.根据't'的数量算eight的数量, e:1 g:1 h:1 i:1
107    temp        =    mci['t']    ;
108    numCount[8] =    temp        ;
109    mci['t']    =    0            ;
110    mci['e']    -=    temp        ;
111    mci['g']    -=    temp        ;
112    mci['h']    -=    temp        ;
113    mci['i']    -=    temp        ;
114
115
116    // 10.根据'e'的数量算nine的数量, i:1 n:2
117    numCount[9] = mci['e']    ;
118
119    for(i = 0 ; i < 10; i++)
120    {
121        // s(num,c)
122        if(numCount[i])

```

```

123         {
124             ret_val += string(numCount[i],to_string(i));
125         }
126     }
127
128     return ret_val;
129
130 }
131 };
132
133
134 /*
135 提交报错,原因不明.
136 执行出错信息: terminate called after throwing an instance of
137 'std::length_error'
138 最后执行的输入: "fviefuro"
139 */

```

## 方法二:C++\_array

```

1  class Solution
2  {
3      public:
4          string originalDigits(string s)
5          {
6              int    charCount[10]    = {0,} ; /* z o w t r v x s g i */
7              int    i                = 0    ;
8              string ret_val          ;
9
10             for(i = 0 ; i < s.size();i++)
11             {
12                 switch(s[i])
13                 {
14                     case 'z':
15                         charCount[0]++;
16                         break;
17                     case 'o':
18                         charCount[1]++;
19                         break;
20                     case 'w':
21                         charCount[2]++;
22                         break;
23                     case 't':
24                         charCount[3]++;
25                         break;
26                     case 'r':
27                         charCount[4]++;
28                         break;
29                     case 'v':
30                         charCount[5]++;
31                         break;
32                     case 'x':
33                         charCount[6]++;
34                         break;
35                     case 's':
36                         charCount[7]++;

```

```

37         break;
38         case 'g':
39             charCount[8]++;
40         break;
41         case 'i':
42             charCount[9]++;
43         break;
44     }
45 }
46
47 charCount[3] = charCount[3] - charCount[2] - charCount[8]
48 ;
49 charCount[4] = charCount[4] - charCount[3] - charCount[0]
50 ;
49 charCount[7] = charCount[7] - charCount[6]
50 ;
50 charCount[1] = charCount[1] - charCount[4] - charCount[2]
- charCount[0] ;
51 charCount[5] = charCount[5] - charCount[7]
52 ;
52 charCount[9] = charCount[9] - charCount[8] - charCount[6]
- charCount[5] ;
53
54
55 for(i = 0 ; i < 10;i++)
56 {
57     if(charCount[i])
58     {
59         ret_val += string(charCount[i],(char)(i+'0'));
60     }
61 }
62
63 return ret_val;
64
65
66 }
67 };
68
69 /*
70 执行结果:
71 通过
72 显示详情
73 执行用时 :24 ms, 在所有 cpp 提交中击败了77.94% 的用户
74 内存消耗 :11.8 MB, 在所有 cpp 提交中击败了25.64%的用户
75 */

```