



黑客派

CTCI 系列 --1.1 判断一个字符串中的字符是否唯一 (C 语言)

作者: [undersky](#)

原文链接: <https://hacpai.com/article/1479176134438>

来源网站: [黑客派](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

<p>原题: </p>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
    (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<blockquote>
<p>Implement an algorithm to determine if a string has all unique characters.What if you ca
not use additional data structures?</p>
</blockquote>
<blockquote>
<p>实现一个算法，判断一个字符串中的字符是否是唯一的。不能够使用额外的数据结构。 </p>
</blockquote>
<hr>
<p>首先得确定构成字符串的字符集有多大？是 ASCII 字符，还是只是 26 个字母？根据不同情况，
们的解决方案可能不同。</p>
<p>下面我们以 ASCII 字符为例，ASCII 码字符可以使用一个字节来表示，有效取值范围为 0~127
总共 128 个字符。<br> 解题思路为，定义一个 128 字节大小 char 型数组，数组初始化为 0。遍历
字符串中的字符，当我们定义的数组中该字符对应位置为 1 则代表该字符在字符串中已出现过，即该字
串中的字符不是唯一的；否则将数据中该字符对应位置置 1。完整遍历一遍字符串，知道到达字符串
尾('\0')。</p>
<pre><code class="language-c highlight-chroma"><span class="highlight-kt">int</span>
span class="highlight-nf">check_uniq</span><span class="highlight-p">(</span><span cla
s="highlight-kt">char</span><span class="highlight-o">*</span><span class="highlight-
">str</span><span class="highlight-p">)</span>
<span class="highlight-p">{</span>
    <span class="highlight-kt">char</span><span class="highlight-n">flag</span><span cl
ss="highlight-p">[</span><span class="highlight-mi">128</span><span class="highlight-
">];</span>
    <span class="highlight-kt">int</span><span class="highlight-n">ret</span><span clas
="highlight-o">=</span><span class="highlight-mi">0</span><span class="highlight-p">
</span>
    <span class="highlight-n">memset</span><span class="highlight-p">(</span><span cla
s="highlight-n">flag</span><span class="highlight-p">,</span><span class="highlight-mi"
>0</span><span class="highlight-p">,</span><span class="highlight-mi"
>0</span><span class="highlight-p">,</span><span class="highlight-k">sizeof</span><span <
span class="highlight-p">(</span><span class="highlight-n">flag</span><span class="highl
ght-p">));</span>
    <span class="highlight-k">while</span><span class="highlight-p">(</span><span clas
="highlight-o">*</span><span class="highlight-n">str</span><span class="highlight-o">
+!</span><span class="highlight-sc">'\0'</span><span class="highlight-p">)</span>
    <span class="highlight-p">{</span>
        <span class="highlight-k">if</span><span class="highlight-p">(</span><span class=
highlight-n">flag</span><span class="highlight-p">[</span><span class="highlight-o">*<
span><span class="highlight-n">str</span><span class="highlight-p">])</span>
        <span class="highlight-p">{</span>
            <span class="highlight-n">ret</span><span class="highlight-o">=</span><span c
ass="highlight-mi">1</span><span class="highlight-p">;</span><span class="highlight-c
">{//有重复
</span><span class="highlight-c1"></span>                <span class="highlight-k">break</spa
><span class="highlight-p">;</span>

```

```

<span class="highlight-p"></span>
<span class="highlight-k">else</span>
<span class="highlight-p">{</span>
    <span class="highlight-n">flag</span><span class="highlight-p">[</span><span class="highlight-o">*</span><span class="highlight-n">str</span><span class="highlight-p">]</span>
<span class="highlight-o">=</span><span class="highlight-mi">1</span><span class="highlight-p"></span>
<span class="highlight-p"></span>
    <span class="highlight-p"></span>
<span class="highlight-p"></span>
    <span class="highlight-k">return</span><span class="highlight-n">ret</span><span class="highlight-p"></span>
<span class="highlight-p"></span>
<span class="highlight-p"></span>
</code></pre>

```

<p>该算法时间复杂度为 $O(n)$ 。算法中标志数组使用了 128Byte，我们还可以进一步减少空间占用原来的 1/8。即每个 ASCII 码字符的出现状况使用一个 bit 位来指示。算法如下：</p>

```

<pre><code class="language-c highlight-chroma"><span class="highlight-kt">int</span>
span class="highlight-nf">check_uniq2</span><span class="highlight-p">(</span><span class="highlight-kt">char</span>
<span class="highlight-o">*</span><span class="highlight-n">str</span><span class="highlight-p">)</span>
<span class="highlight-p">{</span>
    <span class="highlight-kt">int</span><span class="highlight-n">flag</span><span class="highlight-p">[</span><span class="highlight-mi">4</span><span class="highlight-p">]</span>
</code>

```

```

    <span class="highlight-kt">int</span><span class="highlight-n">flag</span><span class="highlight-p">[</span><span class="highlight-mi">4</span><span class="highlight-p">]</span>
</code>

```

```

    <span class="highlight-kt">int</span><span class="highlight-n">ret</span><span class="highlight-p">(</span><span class="highlight-o">=</span><span class="highlight-mi">0</span><span class="highlight-p"></span>
</code>

```

```

    <span class="highlight-n">memset</span><span class="highlight-p">(</span><span class="highlight-n">flag</span><span class="highlight-p">,</span><span class="highlight-mi">0</span><span class="highlight-p">,</span><span class="highlight-k">sizeof</span><span class="highlight-p">(</span><span class="highlight-n">flag</span><span class="highlight-p">));</span>

```

```

    <span class="highlight-k">while</span><span class="highlight-p">(</span><span class="highlight-o">*</span><span class="highlight-n">str</span><span class="highlight-p">+</span><span class="highlight-o">!=</span><span class="highlight-sc">'\0'</span><span class="highlight-p">)</span>

```

```

    <span class="highlight-p">{</span>

```

```

        <span class="highlight-k">if</span><span class="highlight-p">(</span><span class="highlight-n">flag</span><span class="highlight-p">[</span><span class="highlight-o">*</span><span class="highlight-n">str</span><span class="highlight-o">/</span><span class="highlight-mi">32</span><span class="highlight-p">]</span><span class="highlight-o">&&</span>
<span class="highlight-o">&&</span><span class="highlight-p">(</span><span class="highlight-o">*</span><span class="highlight-n">str</span><span class="highlight-o">%</span><span class="highlight-mi">32</span>
</code>

```

```

    <span class="highlight-p">{</span>

```

```

        <span class="highlight-n">ret</span><span class="highlight-o">=</span><span class="highlight-mi">1</span><span class="highlight-p"></span>
</code>

```

```

</code><span class="highlight-c1"></span>        <span class="highlight-k">break</span>
</code>

```

```

    <span class="highlight-p">}</span>

```

```

    <span class="highlight-k">else</span>

```

```

    <span class="highlight-p">{</span>

```

```

        <span class="highlight-n">flag</span><span class="highlight-p">[</span><span class="highlight-p">]</span>

```

```

ss="highlight-o">*/span> <span class="highlight-n">str</span> <span class="highlight-o
>/</span> <span class="highlight-mi">32</span><span class="highlight-p">]</span> <sp
n class="highlight-o">|=</span> <span class="highlight-mi">1</span> <span class="highli
ht-o">&lt;&lt;</span> <span class="highlight-p">(</span><span class="highlight-o">*/</s
an><span class="highlight-n">str</span> <span class="highlight-o">%</span> <span clas
="highlight-mi">32</span><span class="highlight-p">);</span>
    <span class="highlight-p">}</span>
    <span class="highlight-p">}</span>
    <span class="highlight-k">return</span> <span class="highlight-n">ret</span><span cl
ss="highlight-p">;</span>
<span class="highlight-p">}</span>
</code></pre>
<p>测试代码如下: </p>
<pre><code class="language-c highlight-chroma"><span class="highlight-kt">int</span>
span class="highlight-nf">main</span><span class="highlight-p">(</span><span class="h
ghlight-kt">void</span><span class="highlight-p">(</span><span class="h
highlight-p">{</span>
    <span class="highlight-kt">char</span> <span class="highlight-n">buf</span><span cla
s="highlight-p">[</span><span class="highlight-mi">100</span><span class="highlight-p
>];</span>
    <span class="highlight-k">while</span> <span class="highlight-p">(</span><span clas
="highlight-mi">1</span><span class="highlight-p">)</span>
    <span class="highlight-p">{</span>
        <span class="highlight-n">memset</span><span class="highlight-p">(</span><span
lass="highlight-n">buf</span><span class="highlight-p">,</span> <span class="highlight
mi">0</span><span class="highlight-p">,</span> <span class="highlight-k">sizeof</span>
<span class="highlight-p">(</span><span class="highlight-n">buf</span><span class="hi
hlight-p">));</span>
        <span class="highlight-n">printf</span><span class="highlight-p">(</span><span cla
s="highlight-s">"please input string:"</span><span class="highlight-p">);</span>
        <span class="highlight-n">scanf</span><span class="highlight-p">(</span><span cla
s="highlight-s">"%s"</span><span class="highlight-p">,</span> <span class="highlight-n
>buf</span><span class="highlight-p">);</span>
        <span class="highlight-k">if</span> <span class="highlight-p">(</span><span class=
highlight-n">check_uniq2</span><span class="highlight-p">(</span><span class="highlig
t-n">buf</span><span class="highlight-p">))</span>
            <span class="highlight-p">{</span>
                <span class="highlight-n">printf</span><span class="highlight-p">(</span><span
lass="highlight-s">"not uniq !</span><span class="highlight-se">\n</span><span class="h
ghlight-s">"</span><span class="highlight-p">);</span>
            <span class="highlight-p">}</span>
            <span class="highlight-k">else</span>
            <span class="highlight-p">{</span>
                <span class="highlight-n">printf</span><span class="highlight-p">(</span><span
lass="highlight-s">"is uniq !</span><span class="highlight-se">\n</span><span class="hi
hlight-s">"</span><span class="highlight-p">);</span>
            <span class="highlight-p">}</span>
        <span class="highlight-p">}</span>
        <span class="highlight-k">return</span> <span class="highlight-mi">0</span><span cl
ss="highlight-p">;</span>
    <span class="highlight-p">}</span>
</code></pre>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>

```

```
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<p>运行结果: </p>
<p></p>
```