关闭

# **写拉车算法**

2019-09-09 09:20:28 \_-Y-\_-Y-\_ 阅读数 10

编辑

原文链接: https://segmentfault.com/a/1190000008484167

Manacher算法, 又叫"马拉车"算法

可以在时间复杂度为O(n)的情况下求解一个字符串的最长回文子串长度的问题。

### 马拉车算法

回文分为偶回文(abba)和奇回文(abcba)。在处理奇偶回文之间有差异,所以用到了一个技巧

在 (每个字符之间和开头) 插入一个无关的字符,

比如说: abbahopxpo 转换为 \$#a#b#b#a#h#o#p#x#p#o# ('\$' 是防止越界)

起初有一个偶回文abba和一个奇回文opxpo,被转换为#a#b#b#a#和#o#p#x#p#o#,长度都转换成了奇数。

## 回文半径数组radius

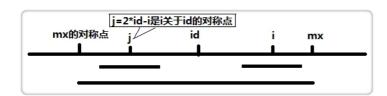
回文半径数组radius是用来记录以每个位置的字符为回文中心求出的回文半径长度,如下图所示

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
s_new[i]	\$	#	a	#	b	#	b	#	a	#	h	#	0	#	р	#	x	#	р	#
radius[i]		1	2	1	2	5	2	1	2	1	2	1	2	1	2	1	4	1	2	1

可以看出, radius[i] - 1正好是原字符串中最长回文串的长度。

#### 求解回文半径数组radius

mx 代表以 id 为中心的最长回文的右边界,也就是mx = id + radius[id]。



如图, mx 到 mx 关于 id 对称点之间是一个回文串,

要是从 j 到 mx 的对称点 之间存在回文串, 那么 i 到 mx 之间也一定会存在回文串

那么以 i 为对称点的回文串的长度最少是 min(radius[j], mx - i)

**并且**i = 2 \* id - i

所以最终优化的是 radius[i] = min(radius[2\*id-i], mx-i)

#### code

```
1 | /*
2
   最长回文
3
   HDU - 3068
4 https://cn.vjudge.net/problem/HDU-3068
   题面:最长回文长度
5
   解法: 马拉车算法
6
7
8
   #include <bits/stdc++.h>
9
   using namespace std;
10
   #define maxn 300005
11
   int radius[maxn];
12
   char s[maxn];
13
   char s_new[maxn];
```

```
14 | int init(){
15
      int len=strlen(s);
16
     s_new[0]='$';
17
     s_new[1]='#';
18
     int j=2;
19
     for(int i=0;i<len;i++){</pre>
20
       s_new[j++]=s[i];
21
       s_new[j++]='#';
22
     s_new[j]='\0';
23
24
    return j;
25 }
26
    int Manacher(){
27
     int len=init();
28
    int ans=-1;
29
    int id;
30
    int mx=0;
31
    for(int i=1;i<len;i++){
32
      if(i<mx) radius[i]=min(radius[2 * id - i], mx - i);</pre>
33
      else radius[i]=1;
      while(s_new[i-radius[i]]==s_new[i+radius[i]]){
34
35
        radius[i]++;
36
       }
37
      if(mx<i+radius[i]){</pre>
38
        id=i;
39
        mx=i+radius[i];
40
41
       ans=max(ans,radius[i]-1);
42
43
     return ans;
44
    }
45
    int main(){
46
     while(~scanf("%s", s)){
47
        printf("%d\n", Manacher());
48
49
    }
50
```

参考: Manacher算法

参考: Manacher算法的详细讲解

有 0 个人打赏

文章最后发布于: 2019-09-09 10:24:13

©2019 CSDN 皮肤主题: 大白 设计师: CSDN官方博客