

[illegible]

Palindromic Tree——回文树【处理一类回文串问题的强力工具】

原创 | pouroul 最后发布于2014-12-23 16:14:22 阅读数 22979 已收藏 展开

今天我们来学习一个神奇的数据结构：Palindromic Tree。中译过来就是——回文树。

那么这个回文树有何功能？

1.求串S的前*i*个不同回文串的个数（两个串长度不同或者长度相同但至少有一个字符不同便是不同串）

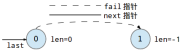
1. 求串S中0-9内本质不同回文串的个数 (两个串长度不同或者长度相同但至少有一个字符不同便是本质不同)
2. 求串S内每一个本质不同回文串出现的次数
3. 求串S内回文串的个数 (其实就是前2结合起来)
4. 求以下标结尾的回文串的个数

那么我们该如何构造回文树?

首先我们定义一些变量。

- 2. **node** 以编号为节点的节点表示的文档树的根 (一个节点表示一个文档字)
- 3. **nodeid** 以编号为节点的节点表示的文档字在两边加括号以后形成的文档字的编号 (与字典树类似)
- 4. **nodeid** 表示节点缺失以后继续不等于自身的节点表示的文档字的最长后缀文档字 (和自动机类似)
- 5. **nodeid** 表示节点表示的文本不同的前缀个数 (建树时节点的不完全, 最后 **nodeid** 函数数一遍以后才呈现的)
- 6. **nodeid** 以节点表示的最长原文文档的最长后缀为原文文档的节点表示的文档字个数
- 7. **nodeid** 指向前添加一个字以后所形成的最长后缀为原文字表示的节点
- 8. **nodeid** 表示添加新的字符 (一开始就设置 $= 0$ (可以是任意一个在串中不会出现的字符))
- 9. **nodeid** 表示添加的节点个数
- 10. **nodeid** 表示添加的字符个数

一开始回文树有两个节点。0表示偶数长度串的根和1表示奇数长度串的根，且 $len[0] = 0$ ， $len[1] = -1$ ， $last = 0$ ， $S[0] = -1$ ， $n = 0$ ， $p = 2$ （添加了节点0、1）。



编号	0	1	2	3	4	5	6	7	8
5	-1								

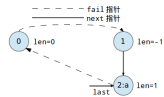
假设现在我们有串 $S = abbaabba$,

首先我们添加第一个字符'a'， $S[++n] = 'a'$ ，然后判断此时 $S[n - \text{len}[\text{last}]]$ 是否等于 $S[n]$ ，即上一个串的位置和新添加的位置是否相同，相同则说明构成回文。否则， $\text{last} = \text{fail}[\text{last}]$ ，此时 $\text{last} = 0$ ，我们发现 $S[1 - 0 - 0] = S[1]$ ，所以 $\text{last} = \text{fail}[\text{last}] = 1$ ，然后我们发现 $S[1 - (-1) - 0] = S[1]$ （即自己等于自己，所以我们让 $\text{len}[1]$ 等于-1可以让这一步更加方便）。

cur等于此时的last (即cur = last = 1)，判断此时next[cur][a]是否已经有后继。如果next[cur][a]没有后继，我们就进行如下的步骤：新建节点（节点数p++，且之后p = 3），并让now等于新节点的编号（now = 2），则len[now] = 2（每个回文节点的长度总是比其最长小于它的节点上的基础上再加上两个相同的字符构成的，所以是+2，同时体现出我们让len[] = 1的优势，一个字符自成一個回文串时回文串的长度为(-1) + 2 = 1），然后我们让fail[now] = next[get_fail(fail[cur])][a]，即得到fail[now]（此时为fail[2] = 0），其中的get_fail函数就是找到第一个使得S[n-len[now]-1] == S[n]的last，然后next[cur][a] = now，即为：

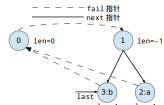
当上面步骤完成后我们让`last = next[cur][c]` (不管`next[cur][a]`是否有后继), 然后`cnt[last] ++`.

此时回文树为下图状态:



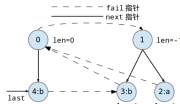
编号	0	1	2	3	4	5	6	7	8
S	-1	a							

现在我们将第二个字符字符'b'到回文树中。



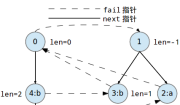
编号	0	1	2	3	4	5	6	7	8
S	-1	a	b						

继续添加第三个字符'b'到回文树中:

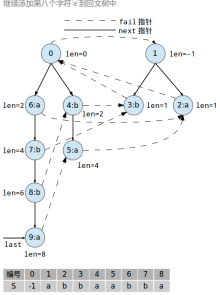
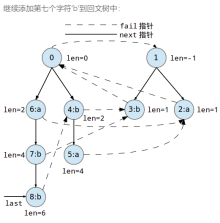
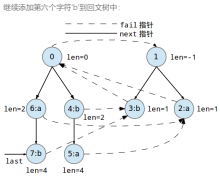
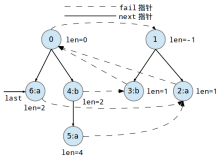


编号	0	1	2	3	4	5	6	7	8
S	-1	a	b	b					

继续添加第四个字符'a'到回文树中:



1674



到此，串S已经完全插入到回文树中了，现在所有的数据如下：

编号	0	1	2	3	4	5	6	7	8	9
char	无	a	b	b	a	a	b	b	a	
fail	1	无	0	0	3	2	2	3	4	5
len	0	1	1	1	2	4	2	4	6	8
cnt	无	无	1	1	1	1	1	1	1	1

然后我们将节点v在fail指针树中将自己的cnt累加给父家，从叶子开始倒着加，最后就能得到串S中出现的每一个本质不同回文串的个数。

构造回文树需要的空间复杂度为O(N*字符集大小)，时间复杂度为O(N*log(字符集大小))，这个时间复杂度比较神奇，如果空间需求太大，可以改成邻接表的形式存储，不过相应的要牺牲一些时间。

总的来说，这是一个很好的算法。

下面贴上我的code

```
1 const int MAXN = 100001;
2 const int M = 26;
3
4 struct Palindromic_Tree {
5     int next[MAXN][M]; //next数组：next指针和转移数组，指针的值为加前缀用添加上一个字符构成
6     int fail[MAXN]; //fail数组：和原回文串树中fail指针指向的节点
7     int cnt[MAXN];
8     int newnode(); //len[]表示节点v表示的回文串的长度
9     int insert(char* s) //将回文串s插入
10     int last; //指向上一个字符添加的节点，为下一个字符
11     int n; //字符个数
12     int p; //当前节点
13
14     int newnode(int len) { //新建节点
15         int i;
16         for (i = 0; i < M; i++) next[i][i] = 0;
17         cnt[i] = 0;
18         len[i] = 0;
19         return i;
20     }
21
22     void init() { //初始化
23         p = 0;
24         newnode(0);
25         newnode(-1);
26         last = 0;
27         n = 0;
28         fail[0] = -1; //从根的一个字符添加的节点，减少特判
29         fail[1] = 1;
30     }
31 }
```


考虑一个只包含小写字母的字符串 s 。我们定义的一个子串的“出现值”为在 s 中的出现次数乘以。 原文 | 来源 | 朱海晨

Codeforces - 17E Palisection 倒置串的回文树 阅读量: 194
题目：给定一个字符串要求找出多少对相交非空的回文子串。思路：直接求相交不好算，我们。 原文 | 来源 | ACWier

Palindromic Tree 回文自动机 回文树 - yzmyh的博客 - CSDN博客

回文树(自动机)练习和总结 [bma204]的专栏 - CSDN博客

Manacher算法：求解最长回文子符串，时间复杂度为O(N) 阅读量: 125+
回文串定义：回文串“是一个正读和反读都一样的字符串。比如“level”或者“noon”等等就是回文串。 原文 | 来源 | yd_wu

回文树(回文树)求解回文串的个数 - MCCME 1750 P04n - CSDN博客

1 - string (本质不同子串数量 本质不同回文子串数量 两个串最高公共自动机 回文自动机) 阅读量: 354
来源：https://ac.nowcoder.com/acm/contest/106A/题解。构造一个串，构造一个串，构造一个串。 原文 | 来源 | juden97

回文树学习小结 阅读量: 450
最近所构造了持久不回文的字符串，研究了一下一直觉得很有趣的回文树，搞了十来道题。现在做。 原文 | 来源 | Ousaf

最长回文子串 (Longest Palindrom Substring) ——三种时间复杂度的解法 阅读量: 1407
题目：给定一个字符串 s ，找出 s 中最长的回文子串。你可以假设 s 的最大长度为1000。来源：Pig_Fly

洛谷 P3649 回文树 回文自动树节点 阅读量: 1444
https://www.luogu.org/problemnew/show/P3649这道题就是一个板子题，回文自动树这个东西。 原文 | 来源 | KSL338

回文树的应用 阅读量: 29
题目：判断一个字符串是否是回文串。回文串是指正读（从左向右）和倒序（从右向左）读都是一样的。 原文 | 来源 | 第三序

树上的回文 阅读量: 194
罗马路2-一棵树。树上有 n 个节点。每个节点有一个小写字母字符串。1号点是树的根。树下有 $n-1$ 个边。 原文 | 来源 | bruceeh

Palindromic Tree 回文自动机 回文树 解决回文串的神器 阅读量: 1469
回文树，也叫回文自动机。是2014年夏天法夫和周发明的一种数据结构。其功能如下：1、求解最长回文串中的本质。 原文 | 来源 | alvin_xf

回文树(Palindromic Tree)+黑科技 学习笔记 阅读量: 2880
回文树Palindromic Tree最基本的回文树在网上有很多资料，在这篇博客中的介绍。本文的重点是。 原文 | 来源 | Yuxuan

【回文树】 阅读量: 15
C语言C/C++ 回文串 为树#include <bits/stdc++.h>using namespace std;typedef。 原文 | 来源 | F0rag

回文树例题 (应用回文串的数量) 阅读量: 36
题目描述You've got a string $s=s_1s_2\ldots s_n$, s_i is of length $|s|$, consisting of lower... 原文 | 来源 | ABBOT

[JzoJ2014]回文串 Hash+树状数组 阅读量: 106
Description给定一个字符串。保证字符串 s 中没有多个回文串。Sample Inputaaaf1 21 31 45a。 原文 | 来源 | vgl_xf

HDU1340: 回文树+马拉车 (本质不同回文子串统计) 阅读量: 703
题目：给定一个字符串，求其本质不同的回文子串个数。如果有14种AA为无效数，请尝试。 原文 | 来源 | 神犇侠

[manacher] 【回文树】回文树进阶小结 阅读量: 171
[manachermanacher]是一个通过以每一个字符串为回文中心的回文串半径的算法。其中，我们约定表示。 原文 | 来源 | wshh15

hhaaccedeeff622：本质不同的回文子串的数量 (manacher+Hash) 阅读量: 204
时间限制1000ms 串的长度1000ms 内存限制5120K 描述给定一个字符串 s ，统计 s 的所有子串。 原文 | 来源 | Mhaoh

回文树小结 阅读量: 239
回文树是一个用来解决回文串相关问题的数据结构。回文树由若干个节点组成，每个节点代表一个。 原文 | 来源 | wang_j

HY58Z 2545 最长回文串 (回文树) 阅读量: 6
2500 最长回文串[4000ms Limit 5120 Secondary Limit 128 MBSubmit 13775Solved 714]Submit[Status]。 原文 | 来源 | wshh15

Palindromic Tree 阅读量: 40
一个很经典的算法。回文串问题不是很难，主要不是理解的问题，知道怎么用就好了。next。 原文 | 来源 | 凡尔赛

[HN省队集训Day2]有趣的字符串问题 回文树+线段树+树状数组 阅读量: 714
题目描述给定一个长度为 n 的字符串 s ，求 s 中所有长度为 k 的子串中，回文子串的数量。回文子串的数量不同回文。 原文 | 来源 | 神犇侠

回文树 (回文) + 例题 阅读量: 1916
说明 Palindromic Tree——回文树【处理一类回文串问题的强力工具】回文树与练习题库首先，回。 原文 | 来源 | 神犇侠

961A Antipalindrome 阅读量: 410
A. Antipalindrome limit per test1 secondmemory limit per test256 megabytestandard in... 原文 | 来源 | mmmmm

大学四年自学成才，这些私藏的实用工具/学习网站我全都整理出来了 阅读量: 1625+
大学四年，感谢身边不可能一直教条书的人了，努力学习，特别感谢中，善于搜索网上的一些资源。 原文 | 来源 | 神犇侠

在中国程序类是有趣吗？ 阅读量: 272+
今年，我也毕业了。为了个给家人们。感谢了家人，感谢了朋友，以及年过15岁的几位老朋友。 原文 | 来源 | 神犇侠

经典Python图像处理讲解 (多图预警) 阅读量: 279+
文章目录How-to讲解一、image模块1.1、打开图像和显示图1.2、创建一个简单的图像1.3、图。 原文 | 来源 | Zax04

为什么所有的都是程序类，基本上不足产品经理的岗位呢？ 阅读量: 103+
相信大家对于产品经理的岗位都很熟悉，也是基本上所有产品经理的岗位。但是为什么？ 原文 | 来源 | 神犇侠

毕业5年，我总结了身边的大佬，总结了他们的学习方法 阅读量: 103+
我写了将近10个大佬，总结了他们的学习方法。原来成功都是有迹可循的。 原文 | 来源 | 神犇侠

推荐10个超棒的神秘的学习网站 阅读量: 2025+
每天都会收到很多读者私信，问我“二叔，有什么推荐的网站吗？”最近在学习，手头的一些。 原文 | 来源 | 神犇侠

Java C语言 Python C++ C# Visual Basic .NET JavaScript PHP SQL Go语言 汇编语言 Assembly language Swift Ruby MATLAB R PL/SQL Perl Visual Basic Objective-C Delphi/Object Pascal Unity3D

没有更多内容了，返回顶部

©2019 CSDN | 回文树 原创 | 设计 | CSDN官方博客