

Manacher算法、扩展KMP

前置知识

无

本期视频会讲述Manacher算法、扩展KMP

下期视频会讲述Manacher算法相关题目

这是讲解Manacher算法、扩展KMP，最清晰的视频

预告

讲解105会讲述字符串哈希的内容

这是解决子串比对问题的利器，比使用KMP算法和next数组更加灵活

Manacher算法、扩展KMP

Manacher算法

求解字符串s的最长回文子串长度只是Manacher算法最微不足道的应用
其中理解回文半径数组有大用处，下节课会非常高频的使用到回文半径数组

扩展KMP

如果理解了Manacher算法，理解扩展KMP是非常容易的

Manacher算法、扩展KMP

理解Manacher算法的重要步骤

- 1, 暴力方法如何寻找最长回文子串
- 2, Manacher扩展串, 可以方便的寻找奇长度、偶长度的回文, 扩展字符可以随意设置, 不会影响计算
- 3, 回文半径和真实回文长度的对应, 真实长度 = $p[i] - 1$
- 4, 扩展回文串结尾下标和真实回文串终止位置的对应, 真实回文串终止位置 = 扩展回文串结尾下标 / 2
- 5, 理解回文半径数组p、理解回文覆盖最右边界r、理解回文中心c, 课上会图解
- 6, Manacher算法的加速过程, 当来到的中心点i, 如何利用p、r、c来进行回文扩展, 课上详细图解
 - a, i没有被r包住, 那么以i为中心直接扩展
 - b, i被r包住, 对称点 $2*c-i$ 的回文半径, 在大回文区域以内, 直接确定 $p[i] = p[2*c-i]$
 - c, i被r包住, 对称点 $2*c-i$ 的回文半径, 在大回文区域以外, 直接确定 $p[i] = r - i$
 - d, i被r包住, 对称点 $2*c-i$ 的回文半径, 撞线大回文区域的边界, 从r之外的位置进行扩展
- 7, Manacher算法的时间复杂度分析, 时间复杂度 $O(n)$
- 8, Manacher算法代码讲解, 理解巧妙的while是如何兼顾四种情况的, 代码中无需四种情况的判断

Manacher算法、扩展KMP

题目1

Manacher算法模版

求字符串s中最长回文子串的长度

测试链接：<https://www.luogu.com.cn/problem/P3805>

尤其是求解回文半径数组，一定要掌握

Manacher算法、扩展KMP

扩展KMP，又称Z算法、Z函数，Z Algorithm

理解了Manacher算法，再去理解扩展KMP会非常容易

理解扩展KMP的重要步骤

- 1, 理解z数组、理解匹配右边界r、理解匹配中心c, 课上会图解
- 2, 理解扩展KMP的过程, 当来到的出发点i, 如何利用z、r、c来进行扩展过程的加速, 课上详细图解
 - a, i没有被r包住, 那么以i为出发点直接扩展
 - b, i被r包住, 但是前点 $i-c$ 的扩展长度, 对应在大扩展区域以内, 直接确定 $z[i] = z[i-c]$
 - c, i被r包住, 但是前点 $i-c$ 的扩展长度, 对应在大扩展区域以外, 直接确定 $z[i] = r - i$
 - d, i被r包住, 但是前点 $i-c$ 的扩展长度, 对应在大扩展区域的边界, 从r之外的位置进行扩展
- 3, 理解了Manacher算法, 可以轻易理解生成z数组的时间复杂度 $O(n)$
- 4, 甚至代码都和Manacher算法高度相似, 巧妙的while兼顾四种情况, 代码中无需四种情况的判断
- 5, 理解了z数组的生成过程, 那么e数组的生成过程同理, 课上会图解

Manacher算法、扩展KMP

题目2

扩展KMP模版，又称Z算法或Z函数

给定两个字符串a、b，求出两个数组

b与b每一个后缀串的最长公共前缀长度，z数组

b与a每一个后缀串的最长公共前缀长度，e数组

计算出要求的两个数组后，输出这两个数组的权值即可

对于一个数组x，i位置的权值定义为： $(i * (x[i] + 1))$

数组权值为所有位置权值的异或和

测试链接：<https://www.luogu.com.cn/problem/P5410>

Manacher算法、扩展KMP

题目3

将单词恢复初始状态所需的最短时间II

给你一个下标从0开始的字符串word和一个整数k

在每一秒，必须执行以下操作

移除word的前k个字符

在word的末尾添加k个任意字符

添加的字符不必和移除的字符相同

返回将word恢复到初始状态所需的最短时间

该时间必须大于零

测试链接：

<https://leetcode.cn/problems/minimum-time-to-revert-word-to-initial-state-ii/>

本题也可以用字符串哈希来解决，将在讲解105讲述字符串哈希，这是解决子串比对问题的利器