

字符串专题

by 黄致焕

kmp

AC自动机

字符串hash

manacher

# 字符串专题

by 黄致焕

2019 年 7 月 28 日

# kmp

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- KMP算法，是由Knuth，Morris，Pratt共同提出的模式匹配算法，其对于任何模式和目标序列，都可以在线性时间内完成匹配查找，而不会发生退化，是一个非常优秀的模式匹配算法。

# kmp

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- KMP算法，是由Knuth，Morris，Pratt共同提出的模式匹配算法，其对于任何模式和目标序列，都可以在线性时间内完成匹配查找，而不会发生退化，是一个非常优秀的模式匹配算法。
- KMP算法的核心是next数组。以模式字符串abcbacacab为例，其next数组为

# kmp

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- KMP算法，是由Knuth，Morris，Pratt共同提出的模式匹配算法，其对于任何模式和目标序列，都可以在线性时间内完成匹配查找，而不会发生退化，是一个非常优秀的模式匹配算法。
- KMP算法的核心是next数组。以模式字符串abcabcacab为例，其next数组为

1	2	3	4	5	6	7	8	9	10
a	b	c	a	b	c	a	c	a	b
0	0	0	1	2	3	4	0	1	2

# kmp

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- KMP算法，是由Knuth，Morris，Pratt共同提出的模式匹配算法，其对于任何模式和目标序列，都可以在线性时间内完成匹配查找，而不会发生退化，是一个非常优秀的模式匹配算法。
- KMP算法的核心是next数组。以模式字符串abcbacacab为例，其next数组为

1	2	3	4	5	6	7	8	9	10
a	b	c	a	b	c	a	c	a	b
0	0	0	1	2	3	4	0	1	2

- 对于字符串S的前i个字符构成的子串，既是它的后缀又是它的前缀的字符串中（它本身除外），最长的长度记作next[i]。

# kmp

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 在不使用kmp算法进行字符串匹配时，我们一般从前向后枚举匹配串的每一个字符，再从前向后枚举模式串的每个字符进行匹配。

# kmp

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 在不使用kmp算法进行字符串匹配时，我们一般从前向后枚举匹配串的每一个字符，再从前向后枚举模式串的每个字符进行匹配。
- 然而这样有很多冗余的比较过程。

# kmp

字符串专题

by 黄致煊

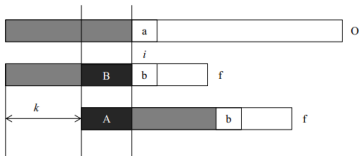
kmp

AC自动机

字符串hash

manacher

- 在不使用kmp算法进行字符串匹配时，我们一般从前向后枚举匹配串的每一个字符，再从前向后枚举模式串的每个字符进行匹配。
- 然而这样有很多冗余的比较过程。





# kmp

字符串专题

by 黄致煊

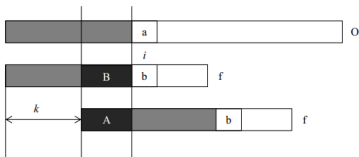
kmp

AC自动机

字符串hash

manacher

- 在不使用kmp算法进行字符串匹配时，我们一般从前向后枚举匹配串的每一个字符，再从前向后枚举模式串的每个字符进行匹配。
- 然而这样有很多冗余的比较过程。



- 根据上图我们发现我们没必要重新从头开始比，而可以直接利用我们预处理的next数组进行跳转。

# kmp

字符串专题

by 黄致煊

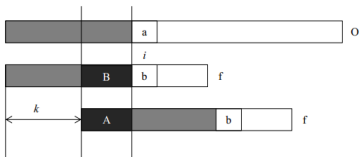
kmp

AC自动机

字符串hash

manacher

- 在不使用kmp算法进行字符串匹配时，我们一般从前向后枚举匹配串的每一个字符，再从前向后枚举模式串的每个字符进行匹配。
- 然而这样有很多冗余的比较过程。



- 根据上图我们发现我们没必要重新从头开始比，而可以直接利用我们预处理的next数组进行跳转。
- 这大大优化了复杂度，使得复杂度从 $O(n^2)$ 优化到了 $O(n)$ 。

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

## ■ 预处理next数组

## ■ 预处理next数组



```
next[0]=-1;
for(i=1;i<=n;i++){
    for(j=next[i-1];j>=0&& s1[j+1]!=s1[i];j=next[j]);
    next[i]=j+1;
}
```

## ■ 进行匹配

## ■ 进行匹配

```
for(i=1,j=0;i<=m;i++){  
    for(;s2[i]!=s1[j+1]&& j>=0;j=next[j]);  
    j++;  
    if(j==n)ans++;  
}
```

# 洛谷P2375 动物园

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 众所周知，在kmp算法中next数组是最长的前缀与后缀相同的长度。
- 现在要求一个num数组表示最长的，不超过一半的，前缀与后缀相同的长度。
- 数据范围 $n \leq 10^6$

# 洛谷P2375 动物园

字符串专题

by 黄致煥

kmp

AC自动机

字符串hash

manacher

- 首先我们可以求出next数组。



# 洛谷P2375 动物园

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们可以求出next数组。
- 在求next数组的同时，我们可以求num数组。首先，显然有  $num_i \leq num_{i-1} + 1$ 。

# 洛谷P2375 动物园

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们可以求出next数组。
- 在求next数组的同时，我们可以求num数组。首先，显然有  $num_i \leq num_{i-1} + 1$ 。
- 所以我们可以直接像求next数组一样利用next数组和  $num_{i-1}$  推出  $num_i$ 。

# 洛谷P2375 动物园

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们可以求出next数组。
- 在求next数组的同时，我们可以求num数组。首先，显然有  $num_i \leq num_{i-1} + 1$ 。
- 所以我们可以直接像求next数组一样利用next数组和  $num_{i-1}$  推出  $num_i$ 。
- 时间复杂度  $O(n)$

# 洛谷P2463 [SDOI2008]Sandy的卡片

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- Sandy和Sue的热衷于收集干脆面中的卡片。
- 然而，Sue收集卡片是因为卡片上漂亮的人物形象，而Sandy则是为了积攒卡片兑换超炫的人物模型。
- 每一张卡片都由一些数字进行标记，第 $i$ 张卡片的序列长度为 $M_i$ ，要想兑换人物模型，首先必须要集够 $N$ 张卡片，对于这 $N$ 张卡片，如果他们都有一个相同的子串长度为 $k$ ，则可以兑换一个等级为 $k$ 的人物模型。相同的定义为：两个子串长度相同且一个串的全部元素加上一个数就会变成另一个串。
- Sandy的卡片数远远小于要求的 $N$ ，于是Sue决定在Sandy的生日将自己的卡片送给Sandy，在Sue的帮助下，Sandy终于集够了 $N$ 张卡片，但是，Sandy并不清楚他可以兑换到哪个等级的人物模型，现在，请你帮助Sandy和Sue，看看他们最高能够得到哪个等级的人物模型。
- 数据范围  $n \leq 1000, m_i \leq 100$

# 洛谷P2463 [SDOI2008]Sandy的卡片

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先可以利用差分把问题变成kmp问题。

# 洛谷P2463 [SDOI2008]Sandy的卡片

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先可以利用差分把问题变成kmp问题。
- 之后我们可以枚举第一个串的所有后缀，把后缀作为模式串和剩下的串进行匹配，可以求出每个后缀在别的串中的最短匹配长度，用这个长度更新答案。

# 洛谷P2463 [SDOI2008]Sandy的卡片

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先可以利用差分把问题变成kmp问题。
- 之后我们可以枚举第一个串的所有后缀，把后缀作为模式串和剩下的串进行匹配，可以求出每个后缀在别的串中的最短匹配长度，用这个长度更新答案。
- 时间复杂度 $O(n * m_i^2)$

# 洛谷P3193 [HNOI2008]GT考试

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 阿申准备报名参加GT考试，准考证号为N位数 $x_1, x_2 \cdots x_n (0 \leq x_i \leq 9)$ ，他不希望准考证号上出现不吉利的数字。
- 他的不吉利数字 $a_1, a_2 \cdots a_m (0 \leq a_i \leq 9)$ 有M位，不出现是指 $x_1, x_2 \cdots x_n$ 中没有恰好一段等于 $a_1, a_2 \cdots a_m$ ， $a_1$ 和 $x_1$ 可以为0。
- 输出方案数对K取模的结果。
- 数据范围 $n \leq 10^9, m \leq 20, K \leq 1000$



# 洛谷P3193 [HNOI2008]GT考试

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 用  $f_{i,j}$  表示前  $i$  个数字和不吉利的数字  $j$  位匹配的方案数。

# 洛谷P3193 [HNOI2008]GT考试

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 用 $f_{i,j}$ 表示前 $i$ 个数字和不吉利的数字 $j$ 位匹配的方案数。
- 转移时只要枚举当前位是多少即可，根据失配情况利用next求出转移方程。

# 洛谷P3193 [HNOI2008]GT考试

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 用 $f_{i,j}$ 表示前 $i$ 个数字和不吉利的数字 $j$ 位匹配的方案数。
- 转移时只要枚举当前位是多少即可，根据失配情况利用next求出转移方程。
- 由于 $n$ 范围较大，所以要利用矩阵乘法优化dp过程。

# 洛谷P3193 [HNOI2008]GT考试

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 用  $f_{i,j}$  表示前  $i$  个数字和不吉利的数字  $j$  位匹配的方案数。
- 转移时只要枚举当前位是多少即可，根据失配情况利用 `next` 求出转移方程。
- 由于  $n$  范围较大，所以要利用矩阵乘法优化 `dp` 过程。
- 时间复杂度  $O(m^3 \log(n))$ 。

# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- AC自动机：Aho-Corasick automation，该算法在1975年产生于贝尔实验室，是著名的多模匹配算法之一。

# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- AC自动机：Aho-Corasick automation，该算法在1975年产生于贝尔实验室，是著名的多模匹配算法之一。
- 很多时候，我们遇到的问题模式串有不止一个，这时候我们就可以使用AC自动机算法来进行匹配。

# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- AC自动机：Aho-Corasick automation，该算法在1975年产生于贝尔实验室，是著名的多模匹配算法之一。
- 很多时候，我们遇到的问题模式串有不止一个，这时候我们就可以使用AC自动机算法来进行匹配。
  - 构造一棵Trie，作为AC自动机的搜索数据结构。

# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- AC自动机：Aho-Corasick automation，该算法在1975年产生于贝尔实验室，是著名的多模匹配算法之一。
- 很多时候，我们遇到的问题模式串有不只一个，这时候我们就可以使用AC自动机算法来进行匹配。
  - 构造一棵Trie，作为AC自动机的搜索数据结构。
  - 构造fail指针，使当前字符失配时跳转到具有最长公共前后缀的字符继续匹配。如同 KMP算法一样，AC自动机在匹配时如果当前字符匹配失败，那么利用fail指针进行跳转。由此可知如果跳转，跳转后的串的前缀，必为跳转前的模式串的后缀并且跳转的新位置的深度（匹配字符个数）一定小于跳之前的节点。所以我们可以利用bfs在Trie上面进行fail指针的求解。



# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 我们给出5个单词，say，she，shr，he，her。给定字符串为yasherhs。问多少个单词在字符串中出现过。

# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 我们给出5个单词，say, she, shr, he, her。给定字符串为yasherhs。问多少个单词在字符串中出现过。
- 首先我们可以建出这些单词的trie树，在每个字符串结尾对应的节点需要特别标记。

# AC自动机

字符串专题

by 黄致煊

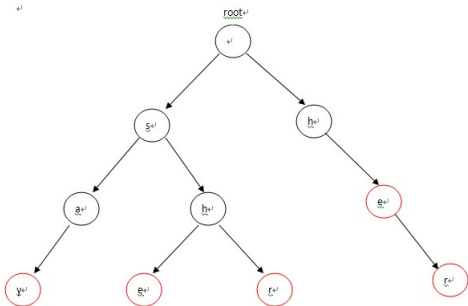
kmp

AC自动机

字符串hash

manacher

- 我们给出5个单词，say, she, shr, he, her。给定字符串为yasherhs。问多少个单词在字符串中出现过。
- 首先我们可以建出这些单词的trie树，在每个字符串结尾对应的节点需要特别标记。



# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 之后我们对于整颗trie树进行一遍dfs，在dfs的同时求出fail数组。每个节点的fail数组可以利用父亲节点的fail数组求得。

# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 之后我们对于整颗trie树进行一遍dfs，在dfs的同时求出fail数组。每个节点的fail数组可以利用父亲节点的fail数组求得。
- 在求完fail数组后，AC自动机就初步建完了。

# AC自动机

字符串专题

by 黄致煊

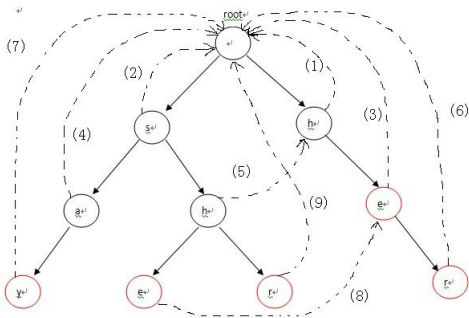
kmp

AC自动机

字符串hash

manacher

- 之后我们对于整颗trie树进行一遍dfs，在dfs的同时求出fail数组。每个节点的fail数组可以利用父亲节点的fail数组求得。
- 在求完fail数组后，AC自动机就初步建完了。



# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 最后我们在AC自动机上跑字符串的时候，我们在失配时，通过fail求出匹配点，用该点更新失配节点对应儿子，这样优化常数。

# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 最后我们在AC自动机上跑字符串的时候，我们在失配时，通过fail求出匹配点，用该点更新失配节点对应儿子，这样优化常数。
- 在匹配方面基本和kmp算法相同，只是将next数组变为fail数组。



# AC自动机

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 最后我们在AC自动机上跑字符串的时候，我们在失配时，通过fail求出匹配点，用该点更新失配节点对应儿子，这样优化常数。
- 在匹配方面基本和kmp算法相同，只是将next数组变为fail数组。
- 时间复杂度 $O(n)$ 。

# hdu 2222 Keywords Search

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 给出 $n$ 个串，然后给一篇文章，问这 $n$ 个串有多少个在文章里面出现过。

# hdu 2222 Keywords Search

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 给出 $n$ 个串，然后给一篇文章，问这 $n$ 个串有多少个在文章里面出现过。
- 数据范围 $n \leq 10000$ 每个串长不超过50，文章长度不超过 $10^6$

# hdu 2222 Keywords Search

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们用这 $n$ 个串建一个AC自动机。之后在这个AC自动机上跑这个文章，将访问过的节点打个标记。

# hdu 2222 Keywords Search

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们用这 $n$ 个串建一个AC自动机。之后在这个AC自动机上跑这个文章，将访问过的节点打个标记。
- 之后枚举每个有标记的点判断这个点是几个字符串的结尾点即可。

# hdu 2222 Keywords Search

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们用这 $n$ 个串建一个AC自动机。之后在这个AC自动机上跑这个文章，将访问过的节点打个标记。
- 之后枚举每个有标记的点判断这个点是几个字符串的结尾点即可。
- 时间复杂度 $O(n)$ 。

# bzoj1030[JSOI2007]文本生成器

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- JSOI交给队员ZYZ一个任务，编制一个称之为文本生成器的电脑软件：该软件的使用者是一些低幼人群，他们现在使用的是GW文本生成器v6版。

# bzoj1030[JSOI2007]文本生成器

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- JSOI交给队员ZYX一个任务，编制一个称之为文本生成器的电脑软件：该软件的使用者是一些低幼人群，他们现在使用的是GW文本生成器v6版。
- 该软件可以随机生成一些文章——总是生成一篇长度固定且完全随机的文章——也就是说，生成的文章中每个字节都是完全随机的。



# bzoj1030[JSOI2007]文本生成器

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- JSOI交给队员ZYX一个任务，编制一个称之为文本生成器的电脑软件：该软件的使用者是一些低幼人群，他们现在使用的是GW文本生成器v6版。
- 该软件可以随机生成一些文章——总是生成一篇长度固定且完全随机的文章——也就是说，生成的文章中每个字节都是完全随机的。
- 如果一篇文章中至少包含使用者们了解的一个单词，那么我们说这篇文章是可读的（我们称文章a包含单词b，当且仅当单词b是文章a的子串）。

# bzoj1030[JSOI2007]文本生成器

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- JSOI交给队员ZYX一个任务，编制一个称之为文本生成器的电脑软件：该软件的使用者是一些低幼人群，他们现在使用的是GW文本生成器v6版。
- 该软件可以随机生成一些文章——总是生成一篇长度固定且完全随机的文章——也就是说，生成的文章中每个字节都是完全随机的。
- 如果一篇文章中至少包含使用者们了解的一个单词，那么我们说这篇文章是可读的（我们称文章a包含单词b，当且仅当单词b是文章a的子串）。
- 但是，即使按照这样的标准，使用者现在使用的GW文本生成器v6版所生成的文章也是几乎完全不可读的。

# bzoj1030[JSOI2007]文本生成器

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- JSOI交给队员ZYX一个任务，编制一个称之为文本生成器的电脑软件：该软件的使用者是一些低幼人群，他们现在使用的是GW文本生成器v6版。
- 该软件可以随机生成一些文章——总是生成一篇长度固定且完全随机的文章——也就是说，生成的文章中每个字节都是完全随机的。
- 如果一篇文章中至少包含使用者们了解的一个单词，那么我们说这篇文章是可读的（我们称文章a包含单词b，当且仅当单词b是文章a的子串）。
- 但是，即使按照这样的标准，使用者现在使用的GW文本生成器v6版所生成的文章也是几乎完全不可读的。
- ZYX需要指出GW文本生成器v6生成的所有文本中可读文本的数量，以便能够成功获得v7更新版。你能帮助他吗？

# bzoj1030[JSOI2007]文本生成器

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- JSOI交给队员ZYX一个任务，编制一个称之为文本生成器的电脑软件：该软件的使用者是一些低幼人群，他们现在使用的是GW文本生成器v6版。
- 该软件可以随机生成一些文章——总是生成一篇长度固定且完全随机的文章——也就是说，生成的文章中每个字节都是完全随机的。
- 如果一篇文章中至少包含使用者们了解的一个单词，那么我们说这篇文章是可读的（我们称文章a包含单词b，当且仅当单词b是文章a的子串）。
- 但是，即使按照这样的标准，使用者现在使用的GW文本生成器v6版所生成的文章也是几乎完全不可读的。
- ZYX需要指出GW文本生成器v6生成的所有文本中可读文本的数量，以便能够成功获得v7更新版。你能帮助他吗？
- $n \leq 60, m \leq 100,$

# bzoj1030[JSOI2007]文本生成器

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 可以在AC自动机上进行dp。

# bzoj1030[JSOI2007]文本生成器

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 可以在AC自动机上进行dp。
- 用 $f[i][j]$ 表示文本第 $i$ 位，在AC自动机上匹配到第 $j$ 个节点，并且之前没有出现过完整单词的方案数。

# bzoj1030[JSOI2007]文本生成器

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 可以在AC自动机上进行dp。
- 用 $f[i][j]$ 表示文本第 $i$ 位，在AC自动机上匹配到第 $j$ 个节点，并且之前没有出现过完整单词的方案数。
- 之后进行DP即可。

# bzoj1030[JSOI2007]文本生成器

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 可以在AC自动机上进行dp。
- 用 $f[i][j]$ 表示文本第 $i$ 位，在AC自动机上匹配到第 $j$ 个节点，并且之前没有出现过完整单词的方案数。
- 之后进行DP即可。
- 时间复杂度 $O(n * m^2)$



# 【bzoj2938】[Poi2000]病毒

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 二进制病毒审查委员会最近发现了如下的规律：某些确定的二进制串是病毒的代码。如果某段代码中不存在任何一段病毒代码，那么我们就称这段代码是安全的。现在委员会已经找出了所有的病毒代码段，试问，是否存在一个无限长的安全的二进制代码。

# 【bzoj2938】[Poi2000]病毒

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 二进制病毒审查委员会最近发现了如下的规律：某些确定的二进制串是病毒的代码。如果某段代码中不存在任何一段病毒代码，那么我们就称这段代码是安全的。现在委员会已经找出了所有的病毒代码段，试问，是否存在一个无限长的安全的二进制代码。
- 例如如果011, 11, 00000为病毒代码段，那么一个可能的无限长安全代码就是010101…。如果01, 11, 000000为病毒代码段，那么就不存在一个无限长的安全代码。

# 【bzoj2938】[Poi2000]病毒

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 二进制病毒审查委员会最近发现了如下的规律：某些确定的二进制串是病毒的代码。如果某段代码中不存在任何一段病毒代码，那么我们就称这段代码是安全的。现在委员会已经找出了所有的病毒代码段，试问，是否存在一个无限长的安全的二进制代码。
- 例如如果011, 11, 00000为病毒代码段，那么一个可能的无限长安全代码就是010101…。如果01, 11, 000000为病毒代码段，那么就不存在一个无限长的安全代码。
  - 读入病毒代码；

# 【bzoj2938】[Poi2000]病毒

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 二进制病毒审查委员会最近发现了如下的规律：某些确定的二进制串是病毒的代码。如果某段代码中不存在任何一段病毒代码，那么我们就称这段代码是安全的。现在委员会已经找出了所有的病毒代码段，试问，是否存在一个无限长的安全的二进制代码。
- 例如如果011, 11, 00000为病毒代码段，那么一个可能的无限长安全代码就是010101…。如果01, 11, 000000为病毒代码段，那么就不存在一个无限长的安全代码。
  - 读入病毒代码；
  - 判断是否存在一个无限长的安全代码；

# 【bzoj2938】[Poi2000]病毒

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 二进制病毒审查委员会最近发现了如下的规律：某些确定的二进制串是病毒的代码。如果某段代码中不存在任何一段病毒代码，那么我们就称这段代码是安全的。现在委员会已经找出了所有的病毒代码段，试问，是否存在一个无限长的安全的二进制代码。
- 例如如果011, 11, 00000为病毒代码段，那么一个可能的无限长安全代码就是010101…。如果01, 11, 000000为病毒代码段，那么就不存在一个无限长的安全代码。
  - 读入病毒代码；
  - 判断是否存在一个无限长的安全代码；
  - 将结果输出

# 【bzoj2938】[Poi2000]病毒

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 二进制病毒审查委员会最近发现了如下的规律：某些确定的二进制串是病毒的代码。如果某段代码中不存在任何一段病毒代码，那么我们就称这段代码是安全的。现在委员会已经找出了所有的病毒代码段，试问，是否存在一个无限长的安全的二进制代码。
- 例如如果011, 11, 00000为病毒代码段，那么一个可能的无限长安全代码就是010101…。如果01, 11, 000000为病毒代码段，那么就不存在一个无限长的安全代码。
  - 读入病毒代码；
  - 判断是否存在一个无限长的安全代码；
  - 将结果输出
- 总长度不超过30000

# 【bzoj2938】[Poi2000]病毒

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们建一个完整版的AC自动机，之后将AC自动机上为病毒代码结尾的节点删去，在剩下的有向图中找是否存在环。

# 【bzoj2938】[Poi2000]病毒

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们建一个完整版的AC自动机，之后将AC自动机上为病毒代码结尾的节点删去，在剩下的有向图中找是否存在环。
- 直接用dfs找环即可。



# 【bzoj2938】[Poi2000]病毒

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们建一个完整版的AC自动机，之后将AC自动机上为病毒代码结尾的节点删去，在剩下的有向图中找是否存在环。
- 直接用dfs找环即可。
- 时间复杂度 $O(30000)$ 。

# 字符串hash

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 有的时候我们需要对两个很长的字符串快速进行比较，而一些匹配算法都不适用时，我们可以使用字符串hash算法。

# 字符串hash

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 有的时候我们需要对两个很长的字符串快速进行比较，而一些匹配算法都不适用时，我们可以使用字符串hash算法。
- 字符串hash方法有很多种，在算法竞赛中最常用的一种就是将字符串变成一定进制的数之后进行取模。这个算法的优点在于支持修改并且速度较快，不容易出现冲突情况。

# 字符串hash

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 有的时候我们需要对两个很长的字符串快速进行比较，而一些匹配算法都不适用时，我们可以使用字符串hash算法。
- 字符串hash方法有很多种，在算法竞赛中最常用的一种就是将字符串变成一定进制的数之后进行取模。这个算法的优点在于支持修改并且速度较快，不容易出现冲突情况。
- 只要预处理进制的次幂即可。

# 【bzoj3555】[Ctsc2014]企鹅QQ

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 小Q是PenguinQQ网站的管理员，他最近在进行一项有趣的研究——哪些账户是同一个人注册的。

# 【bzoj3555】[Ctsc2014]企鹅QQ

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 小Q是PenguinQQ网站的管理员，他最近在进行一项有趣的研究——哪些账户是同一个人注册的。
- 经过长时间的分析，小Q发现同一个人注册的账户名称总是很相似的，例如Penguin1, Penguin2, Penguin3……于是小Q决定先对这种相似的情形进行统计。

# 【bzoj3555】[Ctsc2014]企鹅QQ

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 小Q是PenguinQQ网站的管理员，他最近在进行一项有趣的研究——哪些账户是同一个人注册的。
- 经过长时间的分析，小Q发现同一个人注册的账户名称总是很相似的，例如Penguin1, Penguin2, Penguin3……于是小Q决定先对这种相似的情形进行统计。
- 小Q定义，若两个账户名称是相似的，当且仅当这两个字符串等长且恰好只有一位不同。例如“Penguin1”和“Penguin2”是相似的，但“Penguin1”和“2Penguin”不是相似的。

# 【bzoj3555】[Ctsc2014]企鹅QQ

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 小Q是PenguinQQ网站的管理员，他最近在进行一项有趣的研究——哪些账户是同一个人注册的。
- 经过长时间的分析，小Q发现同一个人注册的账户名称总是很相似的，例如Penguin1, Penguin2, Penguin3……于是小Q决定先对这种相似的情形进行统计。
- 小Q定义，若两个账户名称是相似的，当且仅当这两个字符串等长且恰好只有一位不同。例如“Penguin1”和“Penguin2”是相似的，但“Penguin1”和“2Penguin”不是相似的。
- 而小Q想知道，在给定的  $n$  个账户名称中，有多少对是相似的。为了简化你的工作，小Q给你的  $n$  个字符串长度均等于  $n$ ，且只包含大小写字母、数字、下划线以及“@”共64种字符，而且不存在两个相同的账户名称。



# 【bzoj3555】[Ctsc2014]企鹅QQ

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 小Q是PenguinQQ网站的管理员，他最近在进行一项有趣的研究——哪些账户是同一个人注册的。
- 经过长时间的分析，小Q发现同一个人注册的账户名称总是很相似的，例如Penguin1, Penguin2, Penguin3……于是小Q决定先对这种相似的情形进行统计。
- 小Q定义，若两个账户名称是相似的，当且仅当这两个字符串等长且恰好只有一位不同。例如“Penguin1”和“Penguin2”是相似的，但“Penguin1”和“2Penguin”不是相似的。
- 而小Q想知道，在给定的  $n$  个账户名称中，有多少对是相似的。为了简化你的工作，小Q给你的  $n$  个字符串长度均等于  $n$ ，且只包含大小写字母、数字、下划线以及“@”共64种字符，而且不存在两个相同的账户名称。
- 数据范围  $n \leq 30000, L \leq 200, S \leq 64$

# 【bzoj3555】[Ctsc2014]企鹅QQ

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们可以预处理每个字符串每个前缀和后缀的hash值。

# 【bzoj3555】[Ctsc2014]企鹅QQ

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们可以预处理每个字符串每个前缀和后缀的hash值。
- 之后我们枚举字符串的每一位，利用预处理的hash值求出每个字符串删去这位后的值，将这些值排序，统计答案即可。

# 【bzoj3555】[Ctsc2014]企鹅QQ

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们可以预处理每个字符串每个前缀和后缀的hash值。
- 之后我们枚举字符串的每一位，利用预处理的hash值求出每个字符串删去这位后的值，将这些值排序，统计答案即可。
- 时间复杂度 $O(n \log n * L)$

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 对于一个01字符串，如果将这个字符串0和1取反后，再将整个串反过来和原串一样，就称作“反对称”字符串。

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 对于一个01字符串，如果将这个字符串0和1取反后，再将整个串反过来和原串一样，就称作“反对称”字符串。
- 比如00001111和010101就是反对称的，1001就不是。

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 对于一个01字符串，如果将这个字符串0和1取反后，再将整个串反过来和原串一样，就称作“反对称”字符串。
- 比如00001111和010101就是反对称的，1001就不是。
- 现在给出一个长度为N的01字符串，求它有多少个子串是反对称的。

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 对于一个01字符串，如果将这个字符串0和1取反后，再将整个串反过来和原串一样，就称作“反对称”字符串。
- 比如00001111和010101就是反对称的，1001就不是。
- 现在给出一个长度为N的01字符串，求它有多少个子串是反对称的。
- 数据范围 $n \leq 500000$



# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 我们发现串长一定是偶数。假如串 $[l, r]$ 是合法的，那串 $[l + 1, r - 1]$ 一定也是合法的。

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 我们发现串长一定是偶数。假如串 $[l, r]$ 是合法的，那串 $[l + 1, r - 1]$ 一定也是合法的。
- 首先我们需要预处理这个串从前向后的前缀hash值和从后向前的取反前缀hash值，利用这两个我们可以快速求出每一段区间的hash值。

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 我们发现串长一定是偶数。假如串 $[l, r]$ 是合法的，那串 $[l + 1, r - 1]$ 一定也是合法的。
- 首先我们需要预处理这个串从前向后的前缀hash值和从后向前的取反前缀hash值，利用这两个我们可以快速求出每一段区间的hash值。
- 所以我们可以枚举串中间分割线的位置，之后用二分求出两侧反过来相等的最长串串长。

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 我们发现串长一定是偶数。假如串 $[l, r]$ 是合法的，那串 $[l + 1, r - 1]$ 一定也是合法的。
- 首先我们需要预处理这个串从前向后的前缀hash值和从后向前的取反前缀hash值，利用这两个我们可以快速求出每一段区间的hash值。
- 所以我们可以枚举串中间分割线的位置，之后用二分求出两侧反过来相等的最长串串长。
- 时间复杂度 $O(n \log n)$ 。

# 【bzoj1014】[JSOI2008]火星prefix

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 火星最近研究了一种操作：求一个字串两个后缀的公共前缀。

# 【bzoj1014】[JSOI2008]火星人prefix

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 火星人最近研究了一种操作：求一个字串两个后缀的公共前缀。
- 比方说，有这样一个字符串：madamimadam，现在，火星人定义了一个函数 $LCQ(x, y)$ ，表示：该字符串中第 $x$ 个字符开始的字串，与该字符串中第 $y$ 个字符开始的字串，两个字串的公共前缀的长度。

# 【bzoj1014】[JSOI2008]火星人prefix

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 火星人最近研究了一种操作：求一个字串两个后缀的公共前缀。
- 比方说，有这样一个字符串：madamimadam，现在，火星人定义了一个函数 $LCQ(x, y)$ ，表示：该字符串中第 $x$ 个字符开始的子串，与该字符串中第 $y$ 个字符开始的子串，两个子串的公共前缀的长度。
- 比如， $LCQ(1, 7) = 5$ ,  $LCQ(2, 10) = 1$ ,  $LCQ(4, 7) = 0$  在研究 $LCQ$ 函数的过程中，火星人发现了这样的一个关联：如果把该字符串的所有后缀排好序，就可以很快地求出 $LCQ$ 函数的值；

# 【bzoj1014】[JSOI2008]火星人prefix

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 火星人最近研究了一种操作：求一个字串两个后缀的公共前缀。
- 比方说，有这样一个字符串：madamimadam，现在，火星人定义了一个函数 $LCQ(x, y)$ ，表示：该字符串中第 $x$ 个字符开始的字串，与该字符串中第 $y$ 个字符开始的字串，两个字串的公共前缀的长度。
- 比如， $LCQ(1, 7) = 5$ ,  $LCQ(2, 10) = 1$ ,  $LCQ(4, 7) = 0$  在研究 $LCQ$ 函数的过程中，火星人发现了这样的关联：如果把该字符串的所有后缀排好序，就可以很快地求出 $LCQ$ 函数的值；
- 现在问题变难了：可以更改字符串中某一个字符的值，也可以在字符串中的某一个位置插入一个字符，还能询问 $LCQ(x, y)$ 。



# 【bzoj1014】[JSOI2008]火星prefix

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 火星最近研究了一种操作：求一个字串两个后缀的公共前缀。
- 比方说，有这样一个字符串：madamimadam，现在，火星定义了一个函数 $LCQ(x, y)$ ，表示：该字符串中第 $x$ 个字符开始的子串，与该字符串中第 $y$ 个字符开始的子串，两个子串的公共前缀的长度。
- 比如， $LCQ(1, 7) = 5$ ,  $LCQ(2, 10) = 1$ ,  $LCQ(4, 7) = 0$ 。在研究 $LCQ$ 函数的过程中，火星发现了这样的关联：如果把该字符串的所有后缀排好序，就可以很快地求出 $LCQ$ 函数的值；
- 现在问题变难了：可以更改字符串中某一个字符的值，也可以在字符串中的某一个位置插入一个字符，还能询问 $LCQ(x, y)$ 。
- 数据范围 $m, L \leq 10^5$

# 【bzoj1014】[JSOI2008]火星prefix

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 由于有修改操作，所以我们可以用平衡树来维护整个字符串。

# 【bzoj1014】[JSOI2008]火星prefix

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 由于有修改操作，所以我们可以用平衡树来维护整个字符串。
- 在比较时，我们可以先二分答案，之后在平衡树上求出hash值来进行比较。

# 【bzoj1014】[JSOI2008]火星prefix

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 由于有修改操作，所以我们可以用平衡树来维护整个字符串。
- 在比较时，我们可以先二分答案，之后在平衡树上求出hash值来进行比较。
- 时间复杂度 $O(n \log^2 n)$

# 【野鸡题】01串

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 有一个长度为 $n$ 的01串，还有 $m$ 组询问，每次你可以翻转一个区间，或是询问一个区间是否是一个回文串。

# 【野鸡题】01串

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 有一个长度为 $n$ 的01串，还有 $m$ 组询问，每次你可以翻转一个区间，或是询问一个区间是否是一个回文串。
- 数据范围 $m, L \leq 10^5$

# 【野鸡题】01串

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 我们可以用一个线段树来维护串的hash值，可以快速求出一个串被取反后的hash值。

# 【野鸡题】01串

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 我们可以用一个线段树来维护串的hash值，可以快速求出一个串被取反后的hash值。
- 对于线段树上每个节点维护从前向后的hash值和从后向前的hash值，比较两个值是否相等即可。



# 【野鸡题】01串

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 我们可以用一个线段树来维护串的hash值，可以快速求出一个串被取反后的hash值。
- 对于线段树上每个节点维护从前向后的hash值和从后向前的hash值，比较两个值是否相等即可。
- 时间复杂度 $O(n \log n)$ 。

# manacher

字符串专题

by 黄致煥

kmp

AC自动机

字符串hash

manacher

- Manacher充分利用了回文的性质，从而达到线性时间。

# manacher

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- Manacher充分利用了回文的性质，从而达到线性时间。
- 首先先加一个小优化，就是在每两个字符（包括头尾）之间加没出现的字符（如\*），这样所有字符串长度就都是奇数了，方便了很多。

# manacher

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- Manacher充分利用了回文的性质，从而达到线性时间。
- 首先先加一个小优化，就是在每两个字符（包括头尾）之间加没出现的字符（如\*），这样所有字符串长度就都是奇数了，方便了很多。
- $abcde \rightarrow *a*b*c*d*e*$

# manacher

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- Manacher充分利用了回文的性质，从而达到线性时间。
- 首先先加一个小优化，就是在每两个字符（包括头尾）之间加没出现的字符（如\*），这样所有字符串长度就都是奇数了，方便了很多。
- $abcde \rightarrow *a*b*c*d*e*$
- 记录 $p_i$ 表示 $i$ 能向两边推（包括 $i$ ）的最大距离，如果能求出 $p$ 数组，那每一个回文串都确定了。

# manacher

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- Manacher充分利用了回文的性质，从而达到线性时间。
- 首先先加一个小优化，就是在每两个字符（包括头尾）之间加没出现的字符（如\*），这样所有字符串长度就都是奇数了，方便了很多。
- $abcde \rightarrow *a*b*c*d*e*$
- 记录 $p_i$ 表示 $i$ 能向两边推（包括 $i$ ）的最大距离，如果能求出 $p$ 数组，那每一个回文串都确定了。
- 我们假设 $p[1 \sim i-1]$ 已经求好了，现在要求 $p[i]$ ：假设之前能达到的最右边为 $R$ ，对应的中点为 $pos$ ， $j$ 是 $i$ 的对称点。

# manacher

字符串专题

by 黄致煥

kmp

AC自动机

字符串hash

manacher

- 第一种情况:  $i + p[j] < R$

# manacher

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 第一种情况:  $i + p[j] < R$





# manacher

字符串专题

by 黄致煊

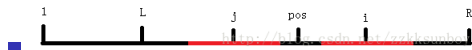
kmp

AC自动机

字符串hash

manacher

- 第一种情况:  $i + p[j] < R$



- 这种情况可以直接求得 $p[i]$ 。

# manacher

字符串专题

by 黄致煊

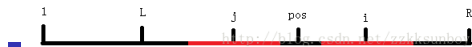
kmp

AC自动机

字符串hash

manacher

- 第一种情况:  $i + p[j] < R$



- 这种情况可以直接求得 $p[i]$ 。
- 第二种情况:  $i + p[j] \geq R$

# manacher

字符串专题

by 黄致煊

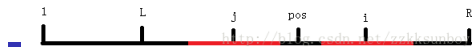
kmp

AC自动机

字符串hash

manacher

- 第一种情况:  $i + p[j] < R$



- 这种情况可以直接求得  $p[i]$ 。

- 第二种情况:  $i + p[j] \geq R$



# manacher

字符串专题

by 黄致煊

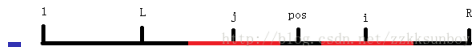
kmp

AC自动机

字符串hash

manacher

- 第一种情况:  $i + p[j] < R$



- 这种情况可以直接求得  $p[i]$ 。

- 第二种情况:  $i + p[j] \geq R$



- 这种情况将  $p[i]$  先设为  $R - i$ , 之后继续增加  $p[i]$ 。

# manacher

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 第一种情况:  $i + p[j] < R$



- 这种情况可以直接求得  $p[i]$ 。

- 第二种情况:  $i + p[j] \geq R$



- 这种情况将  $p[i]$  先设为  $R - i$ , 之后继续增加  $p[i]$ 。

- 由于  $R$  一定是递增的, 所以时间复杂度为  $O(n)$ 。

# manacher

字符串专题

by 黄致煊

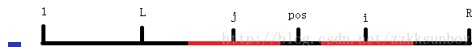
kmp

AC自动机

字符串hash

manacher

- 第一种情况:  $i + p[j] < R$



- 这种情况可以直接求得  $p[i]$ 。

- 第二种情况:  $i + p[j] \geq R$



- 这种情况将  $p[i]$  先设为  $R - i$ , 之后继续增加  $p[i]$ 。
- 由于  $R$  一定是递增的, 所以时间复杂度为  $O(n)$ 。
- 容易发现一个串本质不同的回文串最多有  $n$  个。

# manacher

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

```
1  l=strlen(s1+1);
2  s2[0]='$';
3  s2[ln=1]='#';
4  for(i=1;i<=l;i++)
5      s2[++ln]=s1[i],s2[++ln]='#';
6  for(i=1;i<=ln;i++){
7      if(R>=i)p[i]=min(p[pos*2-i],R-i+1);
8      if(g[i]+i>R){
9          for(;s2[R+1]==s2[i*2-R-1];R++);
10         pos=i;
11         p[i]=R-i+1;
12     }
13 }
```

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 对于一个01字符串，如果将这个字符串0和1取反后，再将整个串反过来和原串一样，就称作“反对称”字符串。



# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 对于一个01字符串，如果将这个字符串0和1取反后，再将整个串反过来和原串一样，就称作“反对称”字符串。
- 比如00001111和010101就是反对称的，1001就不是。

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 对于一个01字符串，如果将这个字符串0和1取反后，再将整个串反过来和原串一样，就称作“反对称”字符串。
- 比如00001111和010101就是反对称的，1001就不是。
- 现在给出一个长度为N的01字符串，求它有多少个子串是反对称的。

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 对于一个01字符串，如果将这个字符串0和1取反后，再将整个串反过来和原串一样，就称作“反对称”字符串。
- 比如00001111和010101就是反对称的，1001就不是。
- 现在给出一个长度为N的01字符串，求它有多少个子串是反对称的。
- 数据范围 $n \leq 500000$

# 【bzoj2084】 [Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 通过观察发现这个问题除了取反之外都满足回文串性质。

# 【bzoj2084】[Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 通过观察发现这个问题除了取反之外都满足回文串性质。
- 所以我们只要在进行比较时取反即可。

# 【bzoj2084】[Poi2010]Antisymmetry

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 通过观察发现这个问题除了取反之外都满足回文串性质。
- 所以我们只要在进行比较时取反即可。
- 时间复杂度 $O(n)$ 。

# 【bzoj2342】[Shoi2011]双倍回文

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

记字符串  $w$  的倒置为  $w^R$ ；例如  $(abcd)^R = dcba$ ， $(abba)^R = abba$ 。

对字符串  $x$ ，如果  $x$  满足  $x^R = x$ ，则称之为回文；例如  $abba$  是一个回文，而  $abcd$  不是。

如果  $x$  能够写成  $ww^Rww^R$  的形式，则称它是一个“双倍回文”。换句话说，若要  $x$  是双倍回文，它的长度必须是 4 的倍数，而且  $x$ 、 $x$  的前半部分、 $x$  的后半部分都要是回文。例如， $abbaabba$  是一个双倍回文，而  $abaaba$  不是，因为它的长度不是 4 的倍数。

$x$  的子串是指在  $x$  中连续的一段字符所组成的字符串。例如  $bc$  是  $abcd$  的子串，而  $ac$  不是。

$x$  的回文子串，就是指满足回文性质的  $x$  的子串。

$x$  的双倍回文子串，就是指满足双倍回文性质的  $x$  的子串。

你的任务是，对于给定的字符串，计算它的最长双倍回文子串的长度。

# 【bzoj2342】[Shoi2011]双倍回文

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

记字符串  $w$  的倒置为  $w^R$ ；例如  $(abcd)^R = dcba$ ， $(abba)^R = abba$ 。

对字符串  $x$ ，如果  $x$  满足  $x^R = x$ ，则称之为回文；例如  $abba$  是一个回文，而  $abcd$  不是。

如果  $x$  能够写成  $ww^Rww^R$  的形式，则称它是一个“双倍回文”。换句话说，若要  $x$  是双倍回文，它的长度必须是 4 的倍数，而且  $x$ 、 $x$  的前半部分、 $x$  的后半部分都要是回文。例如， $abbaabba$  是一个双倍回文，而  $abaaba$  不是，因为它的长度不是 4 的倍数。

$x$  的子串是指在  $x$  中连续的一段字符所组成的字符串。例如  $bc$  是  $abcd$  的子串，而  $ac$  不是。

$x$  的回文子串，就是指满足回文性质的  $x$  的子串。

$x$  的双倍回文子串，就是指满足双倍回文性质的  $x$  的子串。

你的任务是，对于给定的字符串，计算它的最长双倍回文子串的长度。

■ 数据范围  $n \leq 500000$



# 【bzoj2342】[Shoi2011]双倍回文

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们可以枚举双倍回文的中点 $i$ 。

# 【bzoj2342】[Shoi2011]双倍回文

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们可以枚举双倍回文的中点 $i$ 。
- 之后考虑如果 $4 * (i - j)$ 能成为答案，必须满足 $p[j] + j \geq i$ 和 $2 * (i - j) \leq p[i]$ 。

# 【bzoj2342】[Shoi2011]双倍回文

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们可以枚举双倍回文的中点 $i$ 。
- 之后考虑如果 $4 * (i - j)$ 能成为答案，必须满足 $p[j] + j \geq i$ 和 $2 * (i - j) \leq p[i]$ 。
- 所以我们可以用一个set将所有合法的 $j$ 丢进去，实时维护这个set并更新答案即可。

# 【bzoj2342】[Shoi2011]双倍回文

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 首先我们可以枚举双倍回文的中点 $i$ 。
- 之后考虑如果 $4 * (i - j)$ 能成为答案，必须满足 $p[j] + j \geq i$ 和 $2 * (i - j) \leq p[i]$ 。
- 所以我们可以用一个set将所有合法的 $j$ 丢进去，实时维护这个set并更新答案即可。
- 时间复杂度 $O(n \log n)$ 。

# 【bzoj3790】神奇项链

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 母亲节就要到了，小 H 准备送给她一个特殊的项链。这个项链可以看作一个用小写字母组成的字符串，每个小写字母表示一种颜色。

# 【bzoj3790】神奇项链

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 母亲节就要到了，小 H 准备送给她一个特殊的项链。这个项链可以看作一个用小写字母组成的字符串，每个小写字母表示一种颜色。
- 为了制作这个项链，小 H 购买了两个机器。第一个机器可以生成所有形式的回文串，第二个机器可以把两个回文串连接起来，而且第二个机器还有一个特殊的性质：假如一个字符串的后缀和一个字符串的前缀是完全相同的，那么可以将这个重复部分重叠。

# 【bzoj3790】神奇项链

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 母亲节就要到了，小 H 准备送给她一个特殊的项链。这个项链可以看作一个用小写字母组成的字符串，每个小写字母表示一种颜色。
- 为了制作这个项链，小 H 购买了两个机器。第一个机器可以生成所有形式的回文串，第二个机器可以把两个回文串连接起来，而且第二个机器还有一个特殊的性质：假如一个字符串的后缀和一个字符串的前缀是完全相同的，那么可以将这个重复部分重叠。
- 例如：aba和aca连接起来，可以生成串abaaca或 abaca。现在给出目标项链的样式，询问你需要使用第二个机器多少次才能生成这个特殊的项链。

# 【bzoj3790】神奇项链

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 母亲节就要到了，小 H 准备送给她一个特殊的项链。这个项链可以看作一个用小写字母组成的字符串，每个小写字母表示一种颜色。
- 为了制作这个项链，小 H 购买了两个机器。第一个机器可以生成所有形式的回文串，第二个机器可以把两个回文串连接起来，而且第二个机器还有一个特殊的性质：假如一个字符串的后缀和一个字符串的前缀是完全相同的，那么可以将这个重复部分重叠。
- 例如：aba和aca连接起来，可以生成串abaaca或 abaca。现在给出目标项链的样式，询问你需要使用第二个机器多少次才能生成这个特殊的项链。
- 数据范围  $n \leq 10^5$



# 【bzoj3790】神奇项链

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 发现我们只要用所有最长的回文串即可。

# 【bzoj3790】神奇项链

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 发现我们只要用所有最长的回文串即可。
- 问题转化为用一些线段覆盖整个数列。

# 【bzoj3790】神奇项链

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 发现我们只要用所有最长的回文串即可。
- 问题转化为用一些线段覆盖整个数列。
- 用 $f[i]$ 表示数列前 $i$ 位覆盖所需最少的线段树。

# 【bzoj3790】神奇项链

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 发现我们只要用所有最长的回文串即可。
- 问题转化为用一些线段覆盖整个数列。
- 用 $f[i]$ 表示数列前 $i$ 位覆盖所需最少的线段树。
- $f[i]=\min(f[j])+1$ ，前缀和优化dp即可。

# 【bzoj3790】神奇项链

字符串专题

by 黄致煊

kmp

AC自动机

字符串hash

manacher

- 发现我们只要用所有最长的回文串即可。
- 问题转化为用一些线段覆盖整个数列。
- 用 $f[i]$ 表示数列前 $i$ 位覆盖所需最少的线段树。
- $f[i]=\min(f[j])+1$ ，前缀和优化dp即可。
- 时间复杂度 $O(n)$ 。