浅谈回文子串问题

徐毅

江苏省常州高级中学

April 29, 2014

引言

引言

- 字符串问题层出不穷。

引言

- 字符串问题层出不穷。
- 回文子串性质优美，相关问题往往灵活性较强。

引言

- 字符串问题层出不穷。
- 回文子串性质优美，相关问题往往灵活性较强。
- 要求用 $O(n)$ 的时间复杂度求出字符串每一位的回文半径（以该位为中心的回文子串的最大长度的一半）。

引言

- 字符串问题层出不穷。
- 回文子串性质优美，相关问题往往灵活性较强。
- 要求用 $O(n)$ 的时间复杂度求出字符串每一位的回文半径（以该位为中心的回文子串的最大长度的一半）。
- 后缀数组？

引言

- 字符串问题层出不穷。
- 回文子串性质优美，相关问题往往灵活性较强。
- 要求用 $O(n)$ 的时间复杂度求出字符串每一位的回文半径（以该位为中心的回文子串的最大长度的一半）。
- 后缀数组？
- 短小精悍的 Manacher 算法。

Manacher 算法

Manacher 算法

- 奇偶讨论？边界问题？

Manacher 算法

- 奇偶讨论？边界问题？
- 在字符串每一位两侧都添加同一个特殊字符，在首位前和末位后再各自添加一个不同的特殊字符，比如将 abbaba 变为 $#a#b#b#a#b#a#@。

Manacher 算法

- 奇偶讨论？边界问题？
- 在字符串每一位两侧都添加同一个特殊字符，在首位前和末位后再各自添加一个不同的特殊字符，比如将 abbaba 变为 \$#a#b#b#a#b#a#@。
- 通过 Manacher 算法求长度为 n 的新字符串 $S[1\,..\,n]$ 每一位的回文半径 $R[1\,..\,n]$。

Manacher 算法

- 奇偶讨论？边界问题？
- 在字符串每一位两侧都添加同一个特殊字符，在首位前和末位后再各自添加一个不同的特殊字符，比如将 abbaba 变为 \$#a#b#b#a#b#a#@。
- 通过 Manacher 算法求长度为 n 的新字符串 $S[1\,..\,n]$ 每一位的回文半径 $R[1\,..\,n]$。

Table 1: S 和 R 的对应关系

S	#	a	#	b	#	b	#	a	#	b	#	a	#
R	1	2	1	2	5	2	1	4	1	4	1	2	1

Manacher 算法

Manacher 算法

- 添加两个辅助变量 mx 和 p，分别表示已经求过的回文半径覆盖到的最右边界和对应中心的位置。

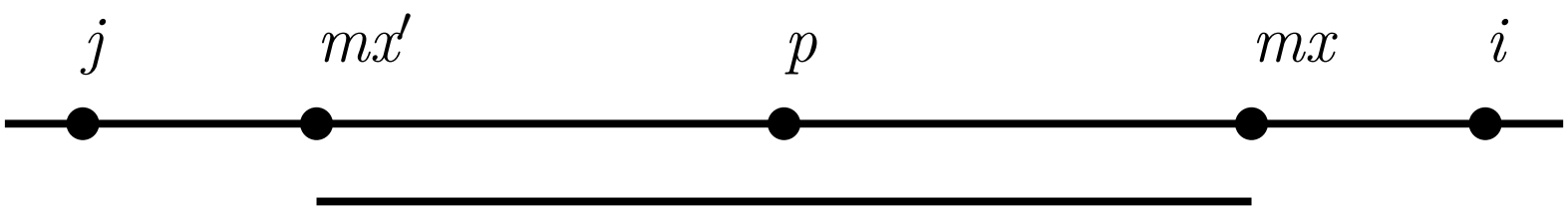
Manacher 算法

- 添加两个辅助变量 mx 和 p，分别表示已经求过的回文半径覆盖到的最右边界和对应中心的位置。
- 计算 $R[i]$ 时，先给它一个下界，令 i 关于 p 的对称点 $j = 2p - i$，分三种情况讨论。

Manacher 算法

- 添加两个辅助变量 mx 和 p，分别表示已经求过的回文半径覆盖到的最右边界和对应中心的位置。
- 计算 $R[i]$ 时，先给它一个下界，令 i 关于 p 的对称点 $j = 2p - i$，分三种情况讨论。
- $mx < i$ 时，只有 $R[i] \geq 1$；

Figure 1: $mx < i$ 的情况（上方线段描述字符串，下方线段描述 $R[p]$）

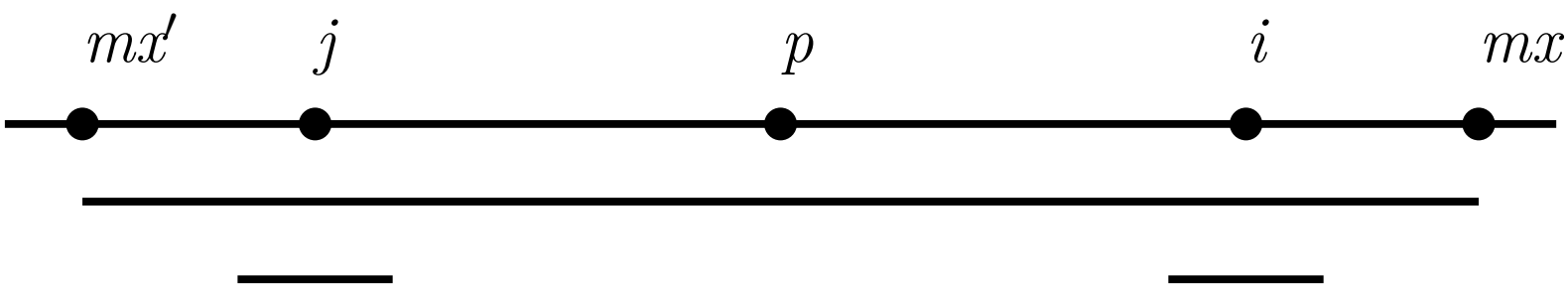


Manacher 算法

Manacher 算法

- $mx - i > R[j]$ 时，以第 j 位为中心的回文子串包含于以第 p 位为中心的回文子串，由于 i 和 j 关于 p 对称，以第 i 位为中心的回文子串必然也包含于以第 p 位为中心的回文子串，故有 $R[i] = R[j]$；

Figure 2: $mx - i > R[j]$ 的情况（上方线段描述字符串，中间线段描述 $R[p]$，左下方线段描述 $R[j]$，右下方线段描述 $R[i]$）

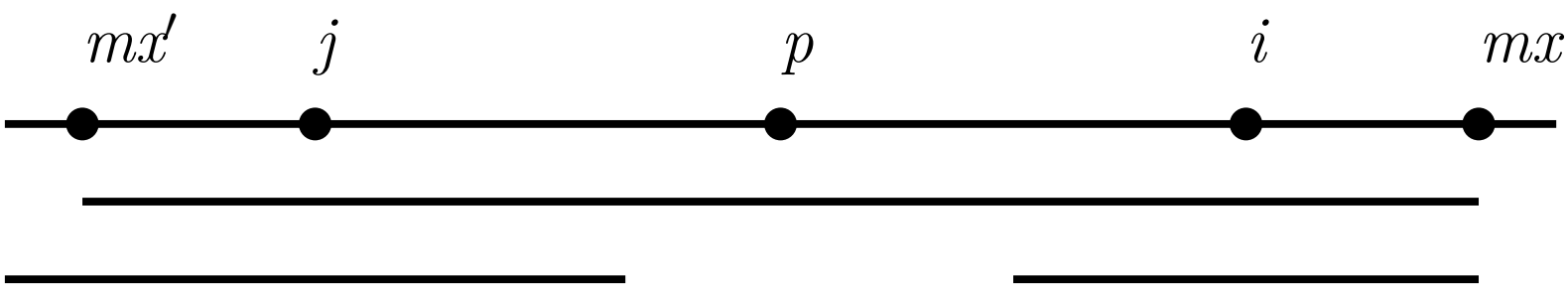


Manacher 算法

Manacher 算法

- $mx - i \leq R[j]$ 时，以第 j 位为中心的回文子串不一定包含于以第 p 位为中心的回文子串，但由于 i 和 j 关于 p 对称，以第 i 位为中心的回文子串向右至少会扩展到 mx 的位置，故有 $R[i] \geq mx - i$。

Figure 3: $mx - i \leq R[j]$ 的情况（上方线段描述字符串，中间线段描述 $R[p]$，左下方线段描述 $R[j]$，右下方线段描述 $R[i]$ 的下界）



Manacher 算法

Manacher 算法

- 超过下界的部分？

Manacher 算法

- 超过下界的部分？
- 逐位匹配。

Manacher 算法

- 超过下界的部分？
- 逐位匹配。
- 实现简单，伪代码参见论文。

Manacher 算法

- 超过下界的部分？
- 逐位匹配。
- 实现简单，伪代码参见论文。
- 时间复杂度为 $O(n)$（逐位匹配成功必然导致 mx 右移，而 mx 右移不超过 n 次）。

Manacher 算法

Manacher 算法

- 解决简单计数问题和求最优字符串问题？

Manacher 算法

- 解决简单计数问题和求最优字符串问题？
- 论文中举了很多这样的例子，由于时间关系不在这里展开。

Manacher 算法

- 解决简单计数问题和求最优字符串问题？
- 论文中举了很多这样的例子，由于时间关系不在这里展开。
- 常见做法是在求出回文半径的基础上，枚举所求字符串的中心或分界点，分析题中的约束条件后结合一些数据结构或是辅助算法来加速单次处理。

例题分析：Palindromic Substring[1]

[1]题目来源：Asia Changchun Regional Contest 2012

例题分析：Palindromic Substring[1]

给出一个长度为 $n(1 \leq n \leq 10^5)$ 的小写字母字符串，进行 $m(1 \leq m \leq 20)$ 次询问，第 i 次询问给每个字母一个 $[0, 25]$ 的权值，求权值第 k_i 小的回文子串（保证存在）的权值（回文子串的权值指将该回文子串的前一半提取出来，每一位用字母对应的权值替代后形成的二十六进制数除以 777777777 的余数）。

[1]题目来源：Asia Changchun Regional Contest 2012

例题分析：Palindromic Substring

例题分析：Palindromic Substring

- 求出每个回文子串的权值后排序？

例题分析：Palindromic Substring

- 求出每个回文子串的权值后排序？
- 回文子串个数？

例题分析：Palindromic Substring

- 求出每个回文子串的权值后排序？
- 回文子串个数？
- **本质相同**的回文子串权值相同。

例题分析：Palindromic Substring

- 求出每个回文子串的权值后排序？
- 回文子串个数？
- **本质相同**的回文子串权值相同。
- **本质不同**的回文子串个数？

例题分析：Palindromic Substring

例题分析：Palindromic Substring

Theorem

一个字符串只有 $O(n)$ 个**本质不同**的回文子串。

例题分析：Palindromic Substring

Theorem

一个字符串只有 $O(n)$ 个**本质不同**的回文子串。

Proof.

我们注意到，在 Manacher 算法中，只有 mx 右移时才会产生新的回文子串（否则一定存在对称的回文子串），而 mx 右移不超过 n 次，故一个字符串只有 $O(n)$ 个本质不同的回文子串。 □

例题分析：Palindromic Substring

Theorem

一个字符串只有 $O(n)$ 个**本质不同**的回文子串。

Proof.

我们注意到，在 Manacher 算法中，只有 mx 右移时才会产生新的回文子串（否则一定存在对称的回文子串），而 mx 右移不超过 n 次，故一个字符串只有 $O(n)$ 个本质不同的回文子串。 □

- 用 Manacher 算法在 $O(n)$ 的时间复杂度内得到所有本质不同的回文子串的位置。

例题分析：Palindromic Substring

例题分析：Palindromic Substring

- 求权值第 k 小？

例题分析：Palindromic Substring

- 求权值第 k 小？
- 统计每种回文子串在该字符串中出现的次数。

例题分析：Palindromic Substring

- 求权值第 k 小？
- 统计每种回文子串在该字符串中出现的次数。
- 按权值从小到大排序，依次累加出现次数直到不小于 k，则对应的权值就是答案。

例题分析：Palindromic Substring

例题分析：Palindromic Substring

- 统计子串在字符串中出现的次数？

例题分析：Palindromic Substring

- 统计子串在字符串中出现的次数？
- 构造后缀数组。

例题分析：Palindromic Substring

- 统计子串在字符串中出现的次数？
- 构造后缀数组。
- 每次统计时二分，时间复杂度为 $O(\log n)$。

例题分析：Palindromic Substring

例题分析：Palindromic Substring

- 计算回文子串的权值？

例题分析：Palindromic Substring

- 计算回文子串的权值？
- 权值的计算规则与字符串 Hash 类似。

例题分析：Palindromic Substring

- 计算回文子串的权值？
- 权值的计算规则与字符串 Hash 类似。
- 每次询问都重新用 $O(n)$ 的时间复杂度预处理，从而用 $O(1)$ 的时间复杂度得到一个回文子串的权值。

例题分析：Palindromic Substring

例题分析：Palindromic Substring

- 构造后缀数组，用 Manacher 算法得到所有本质不同的回文子串的位置，并二分求出其出现次数。

例题分析：Palindromic Substring

- 构造后缀数组，用 Manacher 算法得到所有本质不同的回文子串的位置，并二分求出其出现次数。
- 每次询问重新预处理，将每种回文子串按权值从小到大排序，得到答案。

例题分析：Palindromic Substring

- 构造后缀数组，用 Manacher 算法得到所有本质不同的回文子串的位置，并二分求出其出现次数。
- 每次询问重新预处理，将每种回文子串按权值从小到大排序，得到答案。
- 时间复杂度为 $O(mn \log n)$。

总结

总结

- 根据目标和条件将问题分解，逐步分析。

总结

- 根据目标和条件将问题分解，逐步分析。
- 遇到瓶颈时尝试利用 Manacher 算法的流程来证明某种对象的数量级（通常是 $O(n)$ 的）。

总结

- 根据目标和条件将问题分解，逐步分析。
- 遇到瓶颈时尝试利用 Manacher 算法的流程来证明某种对象的数量级（通常是 $O(n)$ 的）。
- 有时还需要得到一些优美的性质（参见论文中的例题七）。

总结

- 根据目标和条件将问题分解，逐步分析。
- 遇到瓶颈时尝试利用 Manacher 算法的流程来证明某种对象的数量级（通常是 $O(n)$ 的）。
- 有时还需要得到一些优美的性质（参见论文中的例题七）。
- 最终结合多种算法和数据结构解决问题。

致谢

致谢

- 感谢 CCF 提供的学习和交流平台。
- 感谢父母的养育之恩。
- 感谢学校的支持和曹文老师的栽培。
- 感谢许多同学在论文完成过程中对我的帮助。
- 感谢大家的聆听。