

第十五次直播课

习题讲解

李嘉政

Dec 2023

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼
- 6 小蓝的神秘日记
- 7 字符串的变换计数
- 8 诗歌双联
- 9 最长回文子串
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼
- 6 小蓝的神秘日记
- 7 字符串的变换计数
- 8 诗歌双联
- 9 最长回文子串
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆

Solution

枚举答案长度 L ，只需要 n/L 次判断是否子串 Hash 值相等即可。时间复杂度 $\mathcal{O}(n \log n)$ 。

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼
- 6 小蓝的神秘日记
- 7 字符串的变换计数
- 8 诗歌双联
- 9 最长回文子串
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆

Solution

注意到答案有二分性，可以 $\mathcal{O}(n)$ 暴力用 Hash 检验是否有等于当前前缀的，时间复杂度 $\mathcal{O}(n \log n)$ 。

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼
- 6 小蓝的神秘日记
- 7 字符串的变换计数
- 8 诗歌双联
- 9 最长回文子串
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆

Solution

等价于求 border 树上的深度之和，直接用 KMP 建出隐性 border 树，隐性 dfs 即可。时间复杂度 $\mathcal{O}(n)$ 。

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配**
- 5 前缀数量大比拼
- 6 小蓝的神秘日记
- 7 字符串的变换计数
- 8 诗歌双联
- 9 最长回文子串
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆

Solution

倍长后 Hash 判断即可。时间复杂度 $\mathcal{O}(n)$ 。
好像现在数据有点问题。

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼**
- 6 小蓝的神秘日记
- 7 字符串的变换计数
- 8 诗歌双联
- 9 最长回文子串
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆

Solution

考虑 KMP 匹配的过程。对于一个前缀出现的位置而言，它的所有 border 也一定出现过。那么其实对于一个位置而言，只用求出最长能匹配上的前缀即可。这是利用 KMP 的失配容易做到的。时间复杂度 $\mathcal{O}(n)$ 。

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼
- 6 小蓝的神秘日记**
- 7 字符串的变换计数
- 8 诗歌双联
- 9 最长回文子串
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆

Solution

注意到答案具有二分性，判断两个子串一样等价于 Hash 值一样，而不重叠只需要记录一下子串的末尾位置即可。时间复杂度 $\mathcal{O}(n \log^2 n)$ ，分别是二分 + map。

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼
- 6 小蓝的神秘日记
- 7 字符串的变换计数**
- 8 诗歌双联
- 9 最长回文子串
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆

Solution

等价于本质不同循环串个数。简单做法就是倍长后，Hash 判断即可。
时间复杂度 $\mathcal{O}(n \log n)$ 。

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼
- 6 小蓝的神秘日记
- 7 字符串的变换计数
- 8 诗歌双联**
- 9 最长回文子串
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆

Solution

只要求出 t 的前缀匹配上的部分和后缀匹配上的部分不交即可，那么则要求前缀足够靠前，后缀足够靠后，这件事直接利用单调性扫一遍即可，Hash 判断匹配。时间复杂度 $\mathcal{O}(n)$ 。

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼
- 6 小蓝的神秘日记
- 7 字符串的变换计数
- 8 诗歌双联
- 9 最长回文子串**
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆

Solution

Manacher 本质和 KMP 没有区别。

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼
- 6 小蓝的神秘日记
- 7 字符串的变换计数
- 8 诗歌双联
- 9 最长回文子串
- 10 依依的瓶中信**
- 11 小蓝的神秘图书馆

Solution

插入 Trie 的时候同时判断匹配长度，正反各走一遍即可知道所有匹配情况。时间复杂度 $\mathcal{O}(\sum |S_i|)$ 。

Table of Contents

- 1 boarder
- 2 幸运字符串
- 3 你也喜欢幸运字符串吗？
- 4 契合匹配
- 5 前缀数量大比拼
- 6 小蓝的神秘日记
- 7 字符串的变换计数
- 8 诗歌双联
- 9 最长回文子串
- 10 依依的瓶中信
- 11 小蓝的神秘图书馆**

Solution

上课现场做。