

# 字符串问题 匹配与计数等种种

Keavil.Zhang

Tsinghua University  
Computer Science and Technology

2016 年 1 月 21 日

# 字符串工具

# 字符串工具

- ▶ Hashing

# 字符串工具

- ▶ Hashing
- ▶ KMP

# 字符串工具

- ▶ Hashing
- ▶ KMP
- ▶ Aho-Corasick Automaton

# 字符串工具

- ▶ Hashing
- ▶ KMP
- ▶ Aho-Corasick Automaton
- ▶ Suffix Array

# 字符串工具

- ▶ Hashing
- ▶ KMP
- ▶ Aho-Corasick Automaton
- ▶ Suffix Array
- ▶ Suffix Automaton

# 字符串工具

- ▶ Hashing
- ▶ KMP
- ▶ Aho-Corasick Automaton
- ▶ Suffix Array
- ▶ Suffix Automaton
- ▶ Suffix Balance Tree



# 字符串工具

- ▶ Hashing
- ▶ KMP
- ▶ Aho-Corasick Automaton
- ▶ Suffix Array
- ▶ Suffix Automaton
- ▶ Suffix Balance Tree
- ▶ And so on ... (Suffix Cactus?)

# Q1: 匹配问题

## Q1: 匹配问题

- ▶ 给出长度为 $n$ 的串 $A$ 和长度为 $m$ 的串 $B$ ，求串 $B$ 在串 $A$ 中出现的次数。

## Q1: 匹配问题

- ▶ 给出长度为 $n$ 的串A和长度为 $m$ 的串B，求串B在串A中出现的次数。
- ▶ KMP。  $O(n + m)$ 。

## Q2: 多串匹配问题

## Q2: 多串匹配问题

- ▶ 给出长度为 $n$ 的串 $A$ 和长度和为 $L$ 的串集合 $B_i$ ，求每个串在串 $A$ 中的出现次数。

## Q2: 多串匹配问题

- ▶ 给出长度为 $n$ 的串 $A$ 和长度和为 $L$ 的串集合 $B_i$ , 求每个串在串 $A$ 中的出现次数。
- ▶ AC自动机。  $O(n + L)$ 。

## Q2: 多串匹配问题

- ▶ 给出长度为 $n$ 的串 $A$ 和长度和为 $L$ 的串集合 $B_i$ ，求每个串在串 $A$ 中的出现次数。
- ▶ AC自动机。  $O(n + L)$ 。
- ▶ 注意最后在fail树上从下往上依次递推一次。



### Q3: 多串匹配问题（在线版）

### Q3: 多串匹配问题（在线版）

- ▶ 给出长度为 $n$ 的串 $A$ ，每次询问一个串 $B_i$ ，求它在串 $A$ 中的出现次数。询问串总长度为 $L$ 。

### Q3: 多串匹配问题（在线版）

- ▶ 给出长度为 $n$ 的串 $A$ ，每次询问一个串 $B_i$ ，求它在串 $A$ 中的出现次数。询问串总长度为 $L$ 。
- ▶ 后缀数组。

### Q3: 多串匹配问题（在线版）

- ▶ 给出长度为 $n$ 的串 $A$ ，每次询问一个串 $B_i$ ，求它在串 $A$ 中的出现次数。询问串总长度为 $L$ 。
- ▶ 后缀数组。
- ▶ 倍增：  $O(n \log n + L \log n)$ 。

### Q3: 多串匹配问题（在线版）

- ▶ 给出长度为 $n$ 的串 $A$ ，每次询问一个串 $B_i$ ，求它在串 $A$ 中的出现次数。询问串总长度为 $L$ 。
- ▶ 后缀数组。
- ▶ 倍增：  $O(n \log n + L \log n)$ 。
- ▶ DC3: 线性。

### Q3: 多串匹配问题（在线版）

### Q3: 多串匹配问题（在线版）

- ▶ 后缀自动机。

### Q3: 多串匹配问题（在线版）

- ▶ 后缀自动机。
- ▶ 后缀自动机中每个结点的`right`集合大小就是它在串中的出现次数。



### Q3: 多串匹配问题（在线版）

- ▶ 后缀自动机。
- ▶ 后缀自动机中每个结点的`right`集合大小就是它在串中的出现次数。
- ▶ 复杂度为线性。

## Q4: 多模板串匹配问题

## Q4: 多模板串匹配问题

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个长度和为 $K$ 的串 $B_j$ ，求每个串 $B_j$ 在所有 $A_i$ 中出现次数之和。

## Q4: 多模板串匹配问题

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个长度和为 $K$ 的串 $B_j$ ，求每个串 $B_j$ 在所有 $A_i$ 中出现次数之和。
- ▶ 处理多串问题的经典方法：链接成一个串，相邻串中间插入一个不在原本字符集中的字符隔开。

## Q4: 多模板串匹配问题

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个长度和为 $K$ 的串 $B_j$ ，求每个串 $B_j$ 在所有 $A_i$ 中出现次数之和。
- ▶ 处理多串问题的经典方法：链接成一个串，相邻串中间插入一个不在原本字符集中的字符隔开。
- ▶ 另一个经典方法：变成字母树（trie）。这种方法会让后缀自动机复杂度变为 $O(L|\Sigma|)$ 。

## Q4: 多模板串匹配问题

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个长度和为 $K$ 的串 $B_j$ ，求每个串 $B_j$ 在所有 $A_i$ 中出现次数之和。
- ▶ 处理多串问题的经典方法：链接成一个串，相邻串中间插入一个不在原本字符集中的字符隔开。
- ▶ 另一个经典方法：变成字母树（trie）。这种方法会让后缀自动机复杂度变为 $O(L|\Sigma|)$ 。
- ▶ 在线版与Q3相同。

## Q5: 区间多模板串匹配问题

## Q5: 区间多模板串匹配问题

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个长度和为 $K$ 的串 $B_i$ ，求每个串 $B_i$ 在 $i$ 到 $j$ 中间的每个 $A_k$ 中出现次数之和。



## Q5: 区间多模板串匹配问题

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个长度和为 $K$ 的串 $B_i$ ，求每个串 $B_i$ 在 $i$ 到 $j$ 中间的每个 $A_k$ 中出现次数之和。
- ▶ 线段树维护串的集合。线段树上每个节点保存维护区间内串组成的后缀自动机。

## Q5: 区间多模板串匹配问题

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个长度和为 $K$ 的串 $B_i$ ，求每个串 $B_i$ 在 $i$ 到 $j$ 中间的每个 $A_k$ 中出现次数之和。
- ▶ 线段树维护串的集合。线段树上每个节点保存维护区间内串组成的后缀自动机。
- ▶ 每次询问的答案是若干个结点的答案之和。

## Q5: 区间多模板串匹配问题

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个长度和为 $K$ 的串 $B_i$ ，求每个串 $B_i$ 在 $i$ 到 $j$ 中间的每个 $A_k$ 中出现次数之和。
- ▶ 线段树维护串的集合。线段树上每个节点保存维护区间内串组成的后缀自动机。
- ▶ 每次询问的答案是若干个结点的答案之和。
- ▶  $O(L \log n + K \log n)$

## Q6: 区间多模板串匹配问题（带修改）

## Q6: 区间多模板串匹配问题（带修改）

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个总长度为 $K$ 的操作串 $B_i$ ，每个操作为询问 $B_i$ 在 $i$ 到 $j$ 中间的每个 $A_k$ 中出现次数之和，或者将 $B_i$ 加入到 $A_{n+1}$ 的位置。

## Q6: 区间多模板串匹配问题（带修改）

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个总长度为 $K$ 的操作串 $B_i$ ，每个操作为询问 $B_i$ 在 $i$ 到 $j$ 中间的每个 $A_k$ 中出现次数之和，或者将 $B_i$ 加入到 $A_{n+1}$ 的位置。
- ▶ 后缀自动机支持在串尾添加字符。添加操作时只需要修改线段树中根到对应叶子路径上的每一个结点的后缀自动机。

## Q6: 区间多模板串匹配问题（带修改）

- ▶ 给定 $n$ 个长度和为 $L$ 的串 $A_i$ 与 $m$ 个总长度为 $K$ 的操作串 $B_i$ ，每个操作为询问 $B_i$ 在 $i$ 到 $j$ 中间的每个 $A_k$ 中出现次数之和，或者将 $B_i$ 加入到 $A_{n+1}$ 的位置。
- ▶ 后缀自动机支持在串尾添加字符。添加操作时只需要修改线段树中根到对应叶子路径上的每一个结点的后缀自动机。
- ▶ 注意动态维护right集合：动态树维护fail树。

## Q7: 多串匹配问题（在线修改版）



## Q7: 多串匹配问题（在线修改版）

- ▶ 给出长度为 $n$ 的串 $A$ ，每次给定一个串 $B_i$ ，求它在串 $A$ 中的出现次数，或者将 $B_i$ 插入到串 $A$ 的最前面。 $B_i$ 总长度为 $L$ 。

## Q7: 多串匹配问题（在线修改版）

- ▶ 给出长度为 $n$ 的串 $A$ ，每次给定一个串 $B_i$ ，求它在串 $A$ 中的出现次数，或者将 $B_i$ 插入到串 $A$ 的最前面。 $B_i$ 总长度为 $L$ 。
- ▶ 后缀平衡树。 $O((L + n) \log(L + n))$ 。

## Q7: 多串匹配问题（在线修改版）

- ▶ 给出长度为 $n$ 的串 $A$ ，每次给定一个串 $B_i$ ，求它在串 $A$ 中的出现次数，或者将 $B_i$ 插入到串 $A$ 的最前面。 $B_i$ 总长度为 $L$ 。
- ▶ 后缀平衡树。 $O((L + n) \log(L + n))$ 。
- ▶ 还可以支持在串尾删除字符。这是后缀自动机做不到的。

# 小结

# 小结

- ▶ 不同的方法能解决不同的问题。

# 小结

- ▶ 不同的方法能解决不同的问题。
- ▶ 有时需要结合其他方法：比如用可持久化平衡树维护后缀平衡树。

## 小结

- ▶ 不同的方法能解决不同的问题。
- ▶ 有时需要结合其他方法：比如用可持久化平衡树维护后缀平衡树。
- ▶ 有时需要配合离线方法：比如对时间分治后再用后缀数据结构维护。

## 小结

- ▶ 不同的方法能解决不同的问题。
- ▶ 有时需要结合其他方法：比如用可持久化平衡树维护后缀平衡树。
- ▶ 有时需要配合离线方法：比如对时间分治后再用后缀数据结构维护。
- ▶ 不算难题，但是需要想清楚和一定的代码能力。



# 例题

## 例题

- ▶ **Hint:** 能用后缀数组解决的问题几乎全部能用后缀自动机解决。

# 弦论TJOI2015

# 弦论TJOI2015

- ▶ 给定一个字符串 $S$ ，求它的 $K$ 小子串。

# 弦论TJOI2015

- ▶ 给定一个字符串 $S$ ，求它的 $K$ 小子串。
- ▶ 两问，不同位置的相同子串算多个，和不同位置的相同子串算一个。

# 弦论TJOI2015

- ▶ 给定一个字符串 $S$ ，求它的 $K$ 小子串。
- ▶ 两问，不同位置的相同子串算多个，和不同位置的相同子串算一个。
- ▶  $|S| \leq 5 \times 10^5$ ， $K \leq 10^9$ 。

# 弦论TJOI2015

# 弦论TJOI2015

- ▶ 后缀自动机经典问题。从初始点开始按字典序DFS。



# 弦论TJOI2015

- ▶ 后缀自动机经典问题。从初始点开始按字典序DFS。
- ▶ 需要记忆从该点DFS还能有多少个串。

# 弦论TJOI2015

- ▶ 后缀自动机经典问题。从初始点开始按字典序DFS。
- ▶ 需要记忆从该点DFS还能有多少个串。
- ▶ 不同位置算一个：每个结点算一个串。

# 弦论TJOI2015

- ▶ 后缀自动机经典问题。从初始点开始按字典序DFS。
- ▶ 需要记忆从该点DFS还能有多少个串。
- ▶ 不同位置算一个：每个结点算一个串。
- ▶ 不同位置算多个：每个结点串的个数为它的right集合大小。

# 弦论TJOI2015

- ▶ 后缀自动机经典问题。从初始点开始按字典序DFS。
- ▶ 需要记忆从该点DFS还能有多少个串。
- ▶ 不同位置算一个：每个结点算一个串。
- ▶ 不同位置算多个：每个结点串的个数为它的right集合大小。
- ▶  $O(|S|)$ 。

# Cyclical Quest codeforces235C

## Cyclical Quest codeforces235C

- ▶ 给一个串 $S$ ，之后给出 $n$ 个询问，每次询问一个串 $x_i$ ，求 $S$ 有多少子串与 $x_i$ 循环同构。

# Cyclical Quest codeforces235C

- ▶ 给一个串 $S$ ，之后给出 $n$ 个询问，每次询问一个串 $x_i$ ，求 $S$ 有多少子串与 $x_i$ 循环同构。
- ▶ 两个串循环同构指它们可以通过若干次旋转操作（将第一个字符放到最后）来变得相同。

## Cyclical Quest codeforces235C

- ▶ 给一个串 $S$ ，之后给出 $n$ 个询问，每次询问一个串 $x_i$ ，求 $S$ 有多少子串与 $x_i$ 循环同构。
- ▶ 两个串循环同构指它们可以通过若干次旋转操作（将第一个字符放到最后）来变得相同。
- ▶  $|S| \leq 10^6$ ， $|x_i| \leq 10^6$ 。



# Cyclical Quest codeforces235C

# Cyclical Quest codeforces235C

- ▶ 构建S的后缀自动机。

# Cyclical Quest codeforces235C

- ▶ 构建S的后缀自动机。
- ▶ 对每个串 $x_i$ ，复制一份接在后面。要求的就是每个长度为 $|x_i|$ 的子串的出现次数之和。

## Cyclical Quest codeforces235C

- ▶ 构建S的后缀自动机。
- ▶ 对每个串 $x_i$ ，复制一份接在后面。要求的就是每个长度为 $|x_i|$ 的子串的出现次数之和。
- ▶ 经典问题。right集合大小之和。

# Cyclical Quest codeforces235C

- ▶ 构建S的后缀自动机。
- ▶ 对每个串 $x_i$ ，复制一份接在后面。要求的就是每个长度为 $|x_i|$ 的子串的出现次数之和。
- ▶ 经典问题。right集合大小之和。
- ▶  $O(|S| + \sigma|x_i|)$ 。

# 差异AHOI2013

# 差异AHOI2013

- ▶ 给一个字符串 $S$ ，令 $T_i$ 表示第 $i$ 个字符开始的后缀，求：

# 差异AHOI2013

- ▶ 给一个字符串S，令  $T_i$  表示第  $i$  个字符开始的后缀，求：
- ▶  $\sigma_{1 \leq i \leq j \leq |S|} \text{len}(T_i) + \text{len}(T_j) - 2 \times \text{lcp}(T_i, T_j)$



# 差异AHOI2013

- ▶ 给一个字符串 $S$ ，令 $T_i$ 表示第 $i$ 个字符开始的后缀，求：
- ▶  $\sigma_{1 \leq i \leq j \leq |S|} \text{len}(T_i) + \text{len}(T_j) - 2 \times \text{lcp}(T_i, T_j)$
- ▶  $S \leq 5 \times 10^5$ 。

# 差异AHOI2013

# 差异AHOI2013

- ▶ 把字符串反过来，就是求所有前缀的两两最长公共后缀。

# 差异AHOI2013

- ▶ 把字符串反过来，就是求所有前缀的两两最长公共后缀。
- ▶ 找到所有前缀在后缀自动机中的结点，它们的最长公共后缀就是在fail树上的LCA。

# 差异AHOI2013

- ▶ 把字符串反过来，就是求所有前缀的两两最长公共后缀。
- ▶ 找到所有前缀在后缀自动机中的结点，它们的最长公共后缀就是在fail树上的LCA。
- ▶ 经典树形DP。

## 差异AHOI2013

- ▶ 把字符串反过来，就是求所有前缀的两两最长公共后缀。
- ▶ 找到所有前缀在后缀自动机中的结点，它们的最长公共后缀就是在fail树上的LCA。
- ▶ 经典树形DP。
- ▶ 注意有特殊情况：如果一个前缀与另一个前缀的最长公共后缀是它本身，此时的长度就无法达到这个点的len。

# 诸神眷顾的幻想乡ZJOI2015

# 诸神眷顾的幻想乡ZJOI2015

- ▶ 一棵 $n$ 个结点的树，叶子不超过20个，每个结点有一个字符。  
求它有多少条本质不同的路径。



# 诸神眷顾的幻想乡ZJOI2015

- ▶ 一棵 $n$ 个结点的树，叶子不超过20个，每个结点有一个字符。求它有多少条本质不同的路径。
- ▶ 两条路径本质不同指它们路径上结点组成的字符串不同。路径是有向的。

# 诸神眷顾的幻想乡ZJOI2015

- ▶ 一棵 $n$ 个结点的树，叶子不超过20个，每个结点有一个字符。求它有多少条本质不同的路径。
- ▶ 两条路径本质不同指它们路径上结点组成的字符串不同。路径是有向的。
- ▶  $n \leq 10^5$ 。

# 诸神眷顾的幻想乡ZJOI2015

# 诸神眷顾的幻想乡ZJOI2015

- ▶ 注意到叶子个数很少，那么从每个叶子开始把树变成有根树（trie）。

# 诸神眷顾的幻想乡ZJOI2015

- ▶ 注意到叶子个数很少，那么从每个叶子开始把树变成有根树（trie）。
- ▶ 把trie都拼在一起，得到一棵大trie。我们要求这棵trie的本质不同的子串个数。

# 诸神眷顾的幻想乡ZJOI2015

- ▶ 注意到叶子个数很少，那么从每个叶子开始把树变成有根树（trie）。
- ▶ 把trie都拼在一起，得到一棵大trie。我们要求这棵trie的本质不同的子串个数。
- ▶ 后缀自动机。

# 诸神眷顾的幻想乡ZJOI2015

- ▶ 注意到叶子个数很少，那么从每个叶子开始把树变成有根树（trie）。
- ▶ 把trie都拼在一起，得到一棵大trie。我们要求这棵trie的本质不同的子串个数。
- ▶ 后缀自动机。
- ▶ 对trie建立后缀自动机和对串建立比较相似，只要把父节点当做last即可。

# 诸神眷顾的幻想乡ZJOI2015

- ▶ 注意到叶子个数很少，那么从每个叶子开始把树变成有根树（trie）。
- ▶ 把trie都拼在一起，得到一棵大trie。我们要求这棵trie的本质不同的子串个数。
- ▶ 后缀自动机。
- ▶ 对trie建立后缀自动机和对串建立比较相似，只要把父节点当做last即可。
- ▶ 也可以对trie建立后缀数组。



# Musical Theme POJ1743

# Musical Theme POJ1743

- ▶ 给出一个串S，求它的最长的子串，使这个子串至少出现了两次，且有两次出现不重叠。

# Musical Theme POJ1743

- ▶ 给出一个串S，求它的最长的子串，使这个子串至少出现了两次，且有两次出现不重叠。
- ▶  $|S| \leq 10^5$ 。

# Musical Theme POJ1743

# Musical Theme POJ1743

- ▶ 后缀数组。

# Musical Theme POJ1743

- ▶ 后缀数组。
- ▶ 二分答案，如果有一段height都不小于 $x$ ，且存在两个位置的sa差不小于 $x$ ，那么 $x$ 可行。

# Musical Theme POJ1743

- ▶ 后缀数组。
- ▶ 二分答案，如果有一段height都不小于 $x$ ，且存在两个位置的sa差不小于 $x$ ，那么 $x$ 可行。
- ▶ 后缀自动机。

# Musical Theme POJ1743

- ▶ 后缀数组。
- ▶ 二分答案，如果有一段height都不小于 $x$ ，且存在两个位置的sa差不小于 $x$ ，那么 $x$ 可行。
- ▶ 后缀自动机。
- ▶ 对每个结点求出right的最小值和最大值，做差后与该结点的len取较小值，就是这个节点的答案。



# Musical Theme POJ1743

- ▶ 后缀数组。
- ▶ 二分答案，如果有一段height都不小于 $x$ ，且存在两个位置的sa差不小于 $x$ ，那么 $x$ 可行。
- ▶ 后缀自动机。
- ▶ 对每个结点求出right的最小值和最大值，做差后与该结点的len取较小值，就是这个节点的答案。
- ▶ 可以看出后缀自动机的理论复杂度更小。实际运行中因为字符集的问题，比后缀数组只快一点点。

# Typewriter (ACM-ICPC 2015 上海网络赛)

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ 有一个字符串 $s$ ，一开始的输出序列为空。

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ 有一个字符串 $s$ ，一开始的输出序列为空。
- ▶ 每次可以输出一个字符，或者从当前输出序列中复制一段粘贴到输出序列的末尾。

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ 有一个字符串 $s$ ，一开始的输出序列为空。
- ▶ 每次可以输出一个字符，或者从当前输出序列中复制一段粘贴到输出序列的末尾。
- ▶ 输出字符 $c$ 的花费为 $v_c$ ，复制长为 $l$ 的一段并粘贴的花费为 $l \times A + 2B$ 。

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ 有一个字符串 $s$ ，一开始的输出序列为空。
- ▶ 每次可以输出一个字符，或者从当前输出序列中复制一段粘贴到输出序列的末尾。
- ▶ 输出字符 $c$ 的花费为 $v_c$ ，复制长为 $l$ 的一段并粘贴的花费为 $l \times A + 2B$ 。
- ▶ 求最少花费输出给定字符串。

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ 有一个字符串 $s$ ，一开始的输出序列为空。
- ▶ 每次可以输出一个字符，或者从当前输出序列中复制一段粘贴到输出序列的末尾。
- ▶ 输出字符 $c$ 的花费为 $v_c$ ，复制长为 $l$ 的一段并粘贴的花费为 $l \times A + 2B$ 。
- ▶ 求最少花费输出给定字符串。
- ▶  $s \leq 10^5$ 。

# Typewriter (ACM-ICPC 2015 上海网络赛)



# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ DP。

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ DP。
- ▶ 输出一个字符的转移比较显然。

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ DP。
- ▶ 输出一个字符的转移比较显然。
- ▶ 考虑复制一段字符的转移。如果当前输出的是  $1 - i$  这一段的字符，那么我们要找的是一个位置  $j$ ，使字符串  $[j + 1, i]$  是字符串  $[1, j]$  的子串，且  $dp_j + A \times (i - j) + 2B$  最小。

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ DP。
- ▶ 输出一个字符的转移比较显然。
- ▶ 考虑复制一段字符的转移。如果当前输出的是  $1 - i$  这一段的字符，那么我们要找的是一个位置  $j$ ，使字符串  $[j + 1, i]$  是字符串  $[1, j]$  的子串，且  $dp_j + A \times (i - j) + 2B$  最小。
- ▶ 考虑两个转移： $j$  和  $k$ ，其中  $j < k$ 。如果某一次  $j$  的转移没有  $k$  的转移优，那么之后都不会有  $k$  的转移优。而如果  $k$  不能转移了，那么  $j$  一定也不能转移了。

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ DP。
- ▶ 输出一个字符的转移比较显然。
- ▶ 考虑复制一段字符的转移。如果当前输出的是  $1 - i$  这一段的字符，那么我们要找的是一个位置  $j$ ，使字符串  $[j + 1, i]$  是字符串  $[1, j]$  的子串，且  $dp_j + A \times (i - j) + 2B$  最小。
- ▶ 考虑两个转移： $j$  和  $k$ ，其中  $j < k$ 。如果某一次  $j$  的转移没有  $k$  的转移优，那么之后都不会有  $k$  的转移优。而如果  $k$  不能转移了，那么  $j$  一定也不能转移了。
- ▶ 单调性优化。

# Typewriter (ACM-ICPC 2015 上海网络赛)

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ 判断字符串  $[j + 1, i]$  是字符串  $[1, j]$  的子串。

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ 判断字符串  $[j + 1, i]$  是字符串  $[1, j]$  的子串。
- ▶ 子串类问题是后缀自动机所能解决的经典问题。



# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ 判断字符串  $[j + 1, i]$  是字符串  $[1, j]$  的子串。
- ▶ 子串类问题是后缀自动机所能解决的经典问题。
- ▶ 一个串是字符串  $[1 - j]$  的子串，即它所在的结点的 **right** 集合最小值不超过  $j$ 。

# Typewriter (ACM-ICPC 2015 上海网络赛)

- ▶ 判断字符串 $[j + 1, i]$ 是字符串 $[1, j]$ 的子串。
- ▶ 子串类问题是后缀自动机所能解决的经典问题。
- ▶ 一个串是字符串 $[1 - j]$ 的子串，即它所在的结点的`right`集合最小值不超过 $j$ 。
- ▶ 从 $[j, i]$ 到 $[j + 1, i]$ 。沿`fail`指针走。

# 珠宝商CTSC2010

# 珠宝商CTSC2010

- ▶ 一棵 $n$ 个结点的无根树，每个结点有一个字符。有一个长度为 $m$ 的字符串 $S$ 。

# 珠宝商CTSC2010

- ▶ 一棵 $n$ 个结点的无根树，每个结点有一个字符。有一个长度为 $m$ 的字符串 $S$ 。
- ▶ 求树中两两结点（有向）之间字符串在 $S$ 中出现次数之和。

# 珠宝商CTSC2010

- ▶ 一棵 $n$ 个结点的无根树，每个结点有一个字符。有一个长度为 $m$ 的字符串 $S$ 。
- ▶ 求树中两两结点（有向）之间字符串在 $S$ 中出现次数之和。
- ▶  $n, m \leq 5 \times 10^4$ 。

# 珠宝商CTSC2010

# 珠宝商CTSC2010

- ▶ 点分治。



# 珠宝商CTSC2010

- ▶ 点分治。
- ▶ 只考虑经过重心的路径：重心的字符 $x$ 一定出现在了 $S$ 中。

# 珠宝商CTSC2010

- ▶ 点分治。
- ▶ 只考虑经过重心的路径：重心的字符 $x$ 一定出现在了 $S$ 中。
- ▶ 考虑一条可能的路径： $[a, x, b]$ ，那么 $[a, x]$ 的right集合是 $x$ 的right集合子集。同理， $[x, b]$ 反串的right集合是 $x$ 在反串中right集合的子集。

# 珠宝商CTSC2010

- ▶ 点分治。
- ▶ 只考虑经过重心的路径：重心的字符 $x$ 一定出现在了 $S$ 中。
- ▶ 考虑一条可能的路径： $[a, x, b]$ ，那么 $[a, x]$ 的right集合是 $x$ 的right集合子集。同理， $[x, b]$ 反串的right集合是 $x$ 在反串中right集合的子集。
- ▶ 从 $x$ 开始DFS，维护当前串在原串和反串后缀树的位置。将两棵后缀树从上向下递推一遍就得到每一个后缀的匹配数量。两棵后缀树对应位置相乘得到答案。

# 珠宝商CTSC2010

- ▶ 点分治。
- ▶ 只考虑经过重心的路径：重心的字符 $x$ 一定出现在了 $S$ 中。
- ▶ 考虑一条可能的路径： $[a, x, b]$ ，那么 $[a, x]$ 的right集合是 $x$ 的right集合子集。同理， $[x, b]$ 反串的right集合是 $x$ 在反串中right集合的子集。
- ▶ 从 $x$ 开始DFS，维护当前串在原串和反串后缀树的位置。将两棵后缀树从上向下递推一遍就得到每一个后缀的匹配数量。两棵后缀树对应位置相乘得到答案。
- ▶ 注意会多算在同一棵子树内的。需要把每棵子树单独计算一遍。

# 珠宝商CTSC2010

# 珠宝商CTSC2010

- ▶ 问题？

# 珠宝商CTSC2010

- ▶ 问题？
- ▶ 每次都要递推整棵后缀树。

# 珠宝商CTSC2010

- ▶ 问题？
- ▶ 每次都要递推整棵后缀树。
- ▶ 解决方案：只计算大小超过 $\sqrt{n}$ 的分治块。



# 珠宝商CTSC2010

- ▶ 问题？
- ▶ 每次都要递推整棵后缀树。
- ▶ 解决方案：只计算大小超过 $\sqrt{n}$ 的分治块。
- ▶ 对于小于 $\sqrt{n}$ 的分治块，暴力每一条路径。

# 珠宝商CTSC2010

- ▶ 问题？
- ▶ 每次都要递推整棵后缀树。
- ▶ 解决方案：只计算大小超过 $\sqrt{n}$ 的分治块。
- ▶ 对于小于 $\sqrt{n}$ 的分治块，暴力每一条路径。
- ▶  $O((n + m)\sqrt{n})$ 。

str FeyatCup1.5

## str FeyatCup1.5

- ▶ 给定两个长度分别为 $n, m$ 的字符串，定义两个字符串相似为它们至多有一位不相同。求这两个字符串的最长相似子串。

## str FeyatCup1.5

- ▶ 给定两个长度分别为 $n, m$ 的字符串，定义两个字符串相似为它们至多有一位不相同。求这两个字符串的最长相似子串。
- ▶  $n, m \leq 10^5$ 。

str FeyatCup1.5

## str FeyatCup1.5

- ▶ 考虑最后的串，中间有一位不同，其余位都相同。

## str FeyatCup1.5

- ▶ 考虑最后的串，中间有一位不同，其余位都相同。
- ▶ 如果知道不匹配位前的串，那么就知道每一次出现位置，将相应的右边后缀取LCP即可。



## str FeyatCup1.5

- ▶ 考虑最后的串，中间有一位不同，其余位都相同。
- ▶ 如果知道不匹配位前的串，那么就知道每一次出现位置，将相应的右边后缀取LCP即可。
- ▶ 不能两两求LCP：只要求字典序相邻的后缀的LCP。

str FeyatCup1.5

## str FeyatCup1.5

- ▶ 将两个串连起来，考虑每一个结点：它的right集合分为两部分，分别在前后两个串内部。

## str FeyatCup1.5

- ▶ 将两个串连起来，考虑每一个结点：它的right集合分为两部分，分别在前后两个串内部。
- ▶ 那么如果能把整个right集合排序，就只要求相邻的属于不同部分的right值+2后位置的后缀的LCP。

## str FeyatCup1.5

- ▶ 将两个串连起来，考虑每一个结点：它的right集合分为两部分，分别在前后两个串内部。
- ▶ 那么如果能把整个right集合排序，就只要求相邻的属于不同部分的right值+2后位置的后缀的LCP。
- ▶ fail树中自底向上依次考虑，平衡树维护。

## str FeyatCup1.5

- ▶ 将两个串连起来，考虑每一个结点：它的right集合分为两部分，分别在前后两个串内部。
- ▶ 那么如果能把整个right集合排序，就只要求相邻的属于不同部分的right值+2后位置的后缀的LCP。
- ▶ fail树中自底向上依次考虑，平衡树维护。
- ▶ 启发式合并：  $O((n + m) \log(n + m))$ 。

# 总结

# 总结

- ▶ 近期的字符串问题，除了打着字符串幌子的其他类型题外，几乎都是后缀自动机的解决范畴。



# 总结

- ▶ 近期的字符串问题，除了打着字符串幌子的其他类型题外，几乎都是后缀自动机的解决范畴。
- ▶ 比较少见的需要深入理解才能使用的数据结构。

# 总结

- ▶ 近期的字符串问题，除了打着字符串幌子的其他类型题外，几乎都是后缀自动机的解决范畴。
- ▶ 比较少见的需要深入理解才能使用的数据结构。
- ▶ 理解以后属于简单题。

# 总结

- ▶ 近期的字符串问题，除了打着字符串幌子的其他类型题外，几乎都是后缀自动机的解决范畴。
- ▶ 比较少见的需要深入理解才能使用的数据结构。
- ▶ 理解以后属于简单题。
- ▶ 谢谢大家，祝省选顺利。