字符串匹配算法

李佳衡

实验舱科学辅导中心

2020 年 8 月 12 日

字符串匹配

给定一个（或多个）总长为 m 的模式串，再给定一个长为 n 的字符串，求模式串在其中的所有出现位置。

字符串匹配

给定一个（或多个）总长为 m 的模式串，再给定一个长为 n 的字符串，求模式串在其中的所有出现位置。

解

枚举出现位置，朴素地判断匹配，时间复杂度 $O(nm)$。

字符串 Hash

将一个字符串较为随机地映射到一个可以快速比较的整数。

解

选定 Hash 种子 base 与模数 M（需要满足 base 与 M 互质，且 $\mathrm{base} > \max(\Sigma)$），将字符串 $S_{0\ldots n-1}$ Hash 为：

$$\left(\sum_{i=0}^{n-1} S_i \cdot \mathrm{base}^{n-i-1}\right) \bmod M$$

字符串 Hash

将一个字符串较为随机地映射到一个可以快速比较的整数。

解

选定 Hash 种子 base 与模数 M（需要满足 base 与 M 互质，且 $\mathrm{base} > \max(\Sigma)$），将字符串 $S_{0\dots n-1}$ Hash 为：

$$\left(\sum_{i=0}^{n-1} S_i \cdot \mathrm{base}^{n-i-1}\right) \bmod M$$

如果我们认为 Hash 结果是均匀随机的，单次冲突概率仅为 $\frac{1}{M}$。

字符串 Hash

将一个字符串较为随机地映射到一个可以快速比较的整数。

解

选定 Hash 种子 base 与模数 M（需要满足 base 与 M 互质，且 $\mathrm{base} > \max(\Sigma)$），将字符串 $S_{0\dots n-1}$ Hash 为：

$$\left(\sum_{i=0}^{n-1} S_i \cdot \mathrm{base}^{n-i-1}\right) \bmod M$$

如果我们认为 Hash 结果是均匀随机的，单次冲突概率仅为 $\frac{1}{M}$。

如果是 n 个字符串同时出现，出现冲突的概率为 $1-\frac{M(M-1)\dots(M-n+1)}{M^n}$。在 $n=10^5$，$M\approx 10^9$ 时，错误概率高达 0.99 以上，无法接受。

字符串 Hash

因此，$M \approx 10^9$ 无法满足我们的要求，我们需要更大的 M，但更大的 M 在乘法计算上较为复杂。

字符串 Hash

因此，$M \approx 10^9$ 无法满足我们的要求，我们需要更大的 M，但更大的 M 在乘法计算上较为复杂。

自然溢出 Hash

直接取 $M = 2^{64}$，用 unsigned long long 直接计算，不用考虑取模。

此时冲突概率不足 10^{-8}，可以接受。

字符串 Hash

因此，$M \approx 10^9$ 无法满足我们的要求，我们需要更大的 M，但更大的 M 在乘法计算上较为复杂。

自然溢出 Hash

直接取 $M = 2^{64}$，用 unsigned long long 直接计算，不用考虑取模。

此时冲突概率不足 10^{-8}，可以接受。

不幸的是，这一做法并非完全随机，可以构造出两个字符串使得 Hash 冲突。

字符串 Hash

因此，$M \approx 10^9$ 无法满足我们的要求，我们需要更大的 M，但更大的 M 在乘法计算上较为复杂。

自然溢出 Hash

直接取 $M = 2^{64}$，用 unsigned long long 直接计算，不用考虑取模。

此时冲突概率不足 10^{-8}，可以接受。

不幸的是，这一做法并非完全随机，可以构造出两个字符串使得 Hash 冲突。

双 Hash

取两个互质的 M_1、M_2 分别计算、一起比较，由中国剩余定理（CRT），这一做法相当于取 $M = \mathrm{lcm}(M_1, M_2)$。

可以做到 $M \approx 10^{18}$，冲突概率不足 10^{-8}，可以接受。

目前没有已知的构造方法保证这一做法冲突。

字符串 Hash

子串 Hash

设 H_i 表示前缀 $S_{0\dots i-1}$ 的 Hash，则有：

$$H_{i+1} = (H_i \cdot \mathrm{base} + S_i) \bmod M$$

字符串 Hash

子串 Hash

设 H_i 表示前缀 $S_{0\ldots i-1}$ 的 Hash，则有：

$$H_{i+1} = (H_i \cdot \mathrm{base} + S_i) \bmod M$$

一个子串 $S_{l\ldots r-1}$ 的 Hash 即为：

$$(H_r - H_l \cdot \mathrm{base}^{r-l}) \bmod M$$

[BZOJ 3555] 企鹅 QQ

给定 n 个长为 m 的两两不同的字符串，求由多少对字符串仅有一个位置不同。

$n \leq 30000$，$m \leq 200$，字符集大小不超过 64（大小写字母、数字、下划线及 "@"）。

[BZOJ 3555] 企鹅 QQ

给定 n 个长为 m 的两两不同的字符串，求由多少对字符串仅有一个位置不同。

$n \leq 30000$，$m \leq 200$，字符集大小不超过 64（大小写字母、数字、下划线及 "@"）。

解

枚举不同的位置，对于每个字符串求出去掉该位置的 Hash 值，然后排序对于相同的 Hash 计算即可。

时间复杂度 $O(mn \log n)$；如果使用 unordered_map 计算，时间复杂度 $O(mn)$。

[BZOJ 1014] 火星人 prefix

对一个字符串 S 维护 m 次操作：

- **询问** 给定 x、y，求分别以 x、y 开始的两个后缀的 LCP；
- **修改** 将第 x 个位置改为字符 d;
- **插入** 在第 x 个字符之后插入字符 d。

$m \le 150000$，$|S| \le 10^5$，询问不超过 1000 次，字符集为小写字母。

[BZOJ 1014] 火星人 prefix

对一个字符串 S 维护 m 次操作：

询问 给定 x、y，求分别以 x、y 开始的两个后缀的 LCP；

修改 将第 x 个位置改为字符 d;

插入 在第 x 个字符之后插入字符 d。

$m \leq 150000$，$|S| \leq 10^5$，询问不超过 1000 次，字符集为小写字母。

解

用 splay 维护这个字符串，并维护区间的 Hash 值。
询问时二分 LCP 长度，用 splay 提取区间 Hash 比较即可。
时间复杂度 $O(m \log |S| + q \log^2 |S|)$，其中 q 表示询问次数。

[UOJ219] 优秀的拆分

将一个字符串 S 被表示为 $\overline{AABB}$ 的形式（允许 $A = B$），称为 S 的一个优秀的拆分。

给定一个长为 n 的字符串 S，求其所有字串优秀拆分的总个数。

对于 95% 的数据，$n \leq 2000$；对于全部数据，$n \leq 30000$。

[UOJ219] 优秀的拆分

将一个字符串 S 被表示为 $\overline{AABB}$ 的形式（允许 $A = B$），称为 S 的一个优秀的拆分。

给定一个长为 n 的字符串 S，求其所有字串优秀拆分的总个数。

对于 95% 的数据，$n \leq 2000$；对于全部数据，$n \leq 30000$。

解一

考虑枚举 $\overline{AA}$ 与 $\overline{BB}$ 的分界。然后只需求出左、右侧有多少形如 $\overline{AA}$ 的前、后缀，答案相乘即可。可以通过枚举前、后缀用 Hash 判断解决。

时间复杂度 $O(n^2)$，可以通过 95% 的数据。

[UOJ219] 优秀的拆分

将一个字符串 S 被表示为 $\overline{AABB}$ 的形式（允许 $A=B$），称为 S 的一个优秀的拆分。

给定一个长为 n 的字符串 S，求其所有字串优秀拆分的总个数。

对于 95% 的数据，$n \le 2000$；对于全部数据，$n \le 30000$。

解二

分别计算每个点向左的 $\overline{AA}$、向右的 $\overline{BB}$ 个数。

假设我们计算 $\overline{AA}$，枚举这部分的长度 $2l$，将字符串划分为 l 个一块，那么这个子串一定经过且仅经过相邻的两块的左端点，求出 LCP、LCS 后可能的右端点为一个区间，差分维护即可。

其中求 LCP 可以二分 Hash 解决，由调和级数时间复杂度 $O(n\log^2 n)$。

[BZOJ 3097] Hash Killer I

构造 n、l 和一个长为 n 的字符串 S，使得 S 中存在两个长为 l 的子串使 64 位无符号整数自然溢出 Hash 一定产生冲突。

要求 $n \leq 10^5$，只包含小写字母。

[BZOJ 3097] Hash Killer I

构造 n、l 和一个长为 n 的字符串 S，使得 S 中存在两个长为 l 的子串使 64 位无符号整数自然溢出 Hash 一定产生冲突。

要求 $n \le 10^5$，只包含小写字母。

解

如果 base 为偶数，只需连续 65 个相同字符即可。

如果 base 为奇数，可以只使用字母 "a" 和 "b"。设 $A_0 =$ "a"，$A_i = A_{i-1} + \overline{A_{i-1}}$，其中 $\overline{X}$ 表示 "a" 与 "b" 互换。

则 $\mathrm{Hash}(A_i) = \mathrm{Hash}(A_{i-1}) \cdot \mathrm{base}^{2^{i-1}} + \mathrm{Hash}(\overline{A_{i-1}})$。

设 $F_i = \mathrm{Hash}(A_i) - \mathrm{Hash}(\overline{A_i})$，$G_i = \mathrm{base}^{2^{i-1}} - 1$，则 $F_i = F_{i-1} \cdot G_i$。

而 $G_i = \mathrm{base}^{2^{i-1}} - 1 = G_{i-1} \cdot (\mathrm{base}^{2^{i-2}} + 1)$，因此 $2G_{i-1} \mid G_i$。

由上可得 $F_{10} = G_0 G_1 \ldots G_{10}$ 是 2^{64} 的倍数，A_{10} 与 $\overline{A_{10}}$ Hash 冲突。

[UOJ 315] 蚯蚓排队

有 n 个 $[1,6]$ 中的整数 a_i，初始时每个整数分别在一个大小为 1 的队列中。维护 m 次操作：

- 给定 i、j，将 a_j 所在队伍排在 a_i 所在队伍之后，合并两个队伍；
- 给定 i，将 a_i 与其后元素分离，将 a_i 所在队伍分成两个队伍；
- 给定数字串 s、整数 k，对于 s 每个长为 k 的子串 t，求 $f(t)$ 的乘积对 998244353 取模的结果。

其中 $f(t)$ 表示所有队列连成的数字串中，有多少个长为 k 的子串恰好为 t。

$n \le 2 \times 10^5$，$m \le 5 \times 10^5$，$k \le 50$，$\sum |s| \le 10^7$，第二种操作次数 $c \le 1000$。

[UOJ 315] 蚯蚓排队

有 n 个 $[1,6]$ 中的整数 a_i，初始时每个整数分别在一个大小为 1 的队列中。维护 m 次操作：合并、分离、查询。

其中 $f(t)$ 表示所有队列连成的数字串中，有多少个长为 k 的子串恰好为 t。

$n \le 2 \times 10^5$，$m \le 5 \times 10^5$，$k \le 50$，$\sum |s| \le 10^7$，第二种操作次数 $c \le 1000$。

解

每次合并、分离时考虑分界处，枚举有影响的所有长度不超过 $\max\{k\}$ 的子串，用 Hash 维护其出现次数。查询时枚举 s 的子串 Hash 查询即可。

时间复杂度不超过 $O(mk^2 + \sum |s|)$。

[UOJ 315] 蚯蚓排队

有 n 个 $[1,6]$ 中的整数 a_i，初始时每个整数分别在一个大小为 1 的队列中。维护 m 次操作：合并、分离、查询。

其中 $f(t)$ 表示所有队列连成的数字串中，有多少个长为 k 的子串恰好为 t。

$n \le 2\times 10^5$，$m \le 5\times 10^5$，$k \le 50$，$\sum |s| \le 10^7$，第二种操作次数 $c \le 1000$。

解

每次合并、分离时考虑分界处，枚举有影响的所有长度不超过 $\max\{k\}$ 的子串，用 Hash 维护其出现次数。查询时枚举 s 的子串 Hash 查询即可。

时间复杂度不超过 $O(mk^2+\sum |s|)$。

事实上，每个长为 k 的子串只会被计算一次，每次分离会影响至多 k^2 个子串，时间复杂度 $O(mk+ck^2+\sum |s|)$。

KMP 算法

更高效地解决字符串匹配问题。

KMP 算法

更高效地解决字符串匹配问题。

Border

T 是 S 的一个 Border，当且仅当 T 同时是 S 的真前、后缀。

Next 数组

Next_i 表示前缀 $S_{0\ldots i}$ 的最长 Border 长度。

KMP 算法

更高效地解决字符串匹配问题。

Border

T 是 S 的一个 Border，当且仅当 T 同时是 S 的真前、后缀。

Next 数组

Next_i 表示前缀 $S_{0\ldots i}$ 的最长 Border 长度。

计算 Next_i 时，从 Next_{i-1} 开始，不断跳 Border，直到后续位置与 S_i 相同，得到的剩余长度即为 Next_i。

KMP 算法

更高效地解决字符串匹配问题。

Border

T 是 S 的一个 Border，当且仅当 T 同时是 S 的真前、后缀。

Next 数组

Next_i 表示前缀 $S_{0\ldots i}$ 的最长 Border 长度。

计算 Next_i 时，从 Next_{i-1} 开始，不断跳 Border，直到后续位置与 S_i 相同，得到的剩余长度即为 Next_i。

由于额外复杂度与每次 Next 减少次数相同，而 Next 总共只增加了 n，因此时间复杂度 $O(n)$。

KMP 算法

更高效地解决字符串匹配问题。

Border

T 是 S 的一个 Border，当且仅当 T 同时是 S 的真前、后缀。

Next 数组

Next_i 表示前缀 $S_{0\ldots i}$ 的最长 Border 长度。

计算 Next_i 时，从 Next_{i-1} 开始，不断跳 Border，直到后续位置与 S_i 相同，得到的剩余长度即为 Next_i。

由于额外复杂度与每次 Next 减少次数相同，而 Next 总共只增加了 n，因此时间复杂度 $O(n)$。

匹配

对于大串的每新一位，在当前模板串的状态下一直跳 Border，直到下一位能够匹配。

时间复杂度分析同上，为 $O(n+m)$。

[POJ 2185] Milking Grid

给定一个 $n \times m$ 的矩阵，求出一个最小子矩阵，使得其多次平铺包含原矩阵。

$n \leq 10000$，$m \leq 75$。

[POJ 2185] Milking Grid

给定一个 $n \times m$ 的矩阵，求出一个最小子矩阵，使得其多次平铺包含原矩阵。

$n \leq 10000$，$m \leq 75$。

解

先把一行看作一个整体，求出一个最小的 k 使得原矩阵可以由 $k \times m$ 的子矩阵平铺包含。

然后对于这个 $k \times m$ 的矩阵，把一列看作一个整体同样再求最小周期即可。

其中最小周期可以用 KMP 求出，时间复杂度 $O(nm)$。

[BZOJ1009] GT 考试

给定一串数 $\overline{A_1 A_2 \ldots A_m}$，求有多少个数 $\overline{X_1 X_2 \ldots X_n}$ 中不包含 A 作为子串（$0 \le A_i, X_i \le 9$）。答案对读入的 k 取模。

$n \le 10^9$，$m \le 20$，$k \le 1000$。

[BZOJ1009] GT 考试

给定一串数 $\overline{A_1A_2\ldots A_m}$，求有多少个数 $\overline{X_1X_2\ldots X_n}$ 中不包含 A 作为子串（$0 \le A_i, X_i \le 9$）。答案对读入的 k 取模。

$n \le 10^9$，$m \le 20$，$k \le 1000$。

解

考虑 KMP 的匹配过程，将其计入状态 DP。即 $f_{i,j}$ 表示 X 的前 i 位通过 KMP 匹配到 j 且之前没有匹配成功过的方案数，转移枚举这一位的字符即可。时间复杂度 $O(nm)$。

[BZOJ1009] GT 考试

给定一串数 $\overline{A_1A_2\dots A_m}$，求有多少个数 $\overline{X_1X_2\dots X_n}$ 中不包含 A 作为子串（$0 \le A_i, X_i \le 9$）。答案对读入的 k 取模。

$n \le 10^9$，$m \le 20$，$k \le 1000$。

解

考虑 KMP 的匹配过程，将其计入状态 DP。即 $f_{i,j}$ 表示 X 的前 i 位通过 KMP 匹配到 j 且之前没有匹配成功过的方案数，转移枚举这一位的字符即可。时间复杂度 $O(nm)$。

实际上，用向量 V_i 表示 f_i，这个 DP 可以通过矩阵乘法优化，时间复杂度 $O(m^3 \log n)$。

[BZOJ 1100] 对称轴 osi

给一个 n 个点的简单多边形，求其有多少条对称轴。

不超过 10 组数据，$n \leq 10^5$，坐标是 $[1, 10^9]$ 中的整数，相邻两条边不在同一直线上。

[BZOJ 1100] 对称轴 osi

给一个 n 个点的简单多边形，求其有多少条对称轴。

不超过 10 组数据，$n \leq 10^5$，坐标是 $[1, 10^9]$ 中的整数，相邻两条边不在同一直线上。

解

记录每条边的信息（长度、夹角），如果其沿某一位对称后与自身相同，则出现一个对称轴。

[BZOJ 1100] 对称轴 osi

给一个 n 个点的简单多边形，求其有多少条对称轴。

不超过 10 组数据，$n \le 10^5$，坐标是 $[1, 10^9]$ 中的整数，相邻两条边不在同一直线上。

解

记录每条边的信息（长度、夹角），如果其沿某一位对称后与自身相同，则出现一个对称轴。

将信息视作字符，将环倍长为链，反串每在环上出现一次，即出现一个对称轴，可以用 KMP 算法匹配。

时间复杂度 $O(n)$。

[UOJ 5] 动物园

给定一个长为 n 的字符串 S，设 Num_i 表示前缀 $S_{0\ldots i}$ 中最长不重叠的 Border 数量。求 Num 数组。

不超过 5 组数据，$n \le 10^6$。

[UOJ 5] 动物园

给定一个长为 n 的字符串 S，设 Num_i 表示前缀 $S_{0\dots i}$ 中最长不重叠的 Border 数量。求 Num 数组。

不超过 5 组数据，$n \leq 10^6$。

解一

先对字符串求出 Next 数组，再求出 Dep 数组表示 Border 的数量。然后我们考虑求出每个前缀最长的不重叠 Border，依旧使用类似 KMP 的方法跳 Border 递推即可。

时间复杂度 $O(n)$。

[UOJ 5] 动物园

给定一个长为 n 的字符串 S，设 Num_i 表示前缀 $S_{0\ldots i}$ 中最长不重叠的 Border 数量。求 Num 数组。

不超过 5 组数据，$n \leq 10^6$。

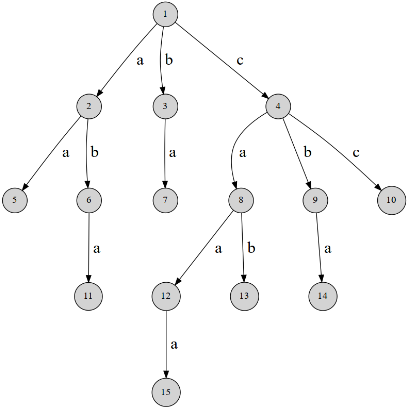
解二

对字符串求出 Next 数组，每次暴力向上跳直到长度不超过一半，这个过程可以用倍增优化。

时间复杂度 $O(n \log n)$。

字典树（Trie）

一棵树上每条边对应一个字符，一个点对应的字符串即为从根到该点的路径。



Trie

[POJ 3764] The xor-longest Path

给定一棵 n 个点、有边权的树，一条路径的权值定义为路径上边权的异或和，求路径权值的最大值。

$n \le 10^6$，边权在 $[0, 2^{31})$ 中。

[POJ 3764] The xor-longest Path

给定一棵 n 个点、有边权的树，一条路径的权值定义为路径上边权的异或和，求路径权值的最大值。

$n \le 10^6$，边权在 $[0, 2^{31})$ 中。

解

求出树上每个点的到根路径权值，则 u、v 间路径的权值即为两者到根权值的异或和，问题变为求 n 个数中两个数异或的最大值，可以用 Trie 解决。

时间复杂度 $O(n \log w)$。

AC 自动机

解决多个模板串（Trie）的字符串匹配问题。

AC 自动机

解决多个模板串（Trie）的字符串匹配问题。

Fail 数组

Fail_i 表示 Trie 上 i 点最长的、在 Trie 上出现过的真后缀对应的结点。

AC 自动机

解决多个模板串（Trie）的字符串匹配问题。

Fail 数组

Fail_i 表示 Trie 上 i 点最长的、在 Trie 上出现过的真后缀对应的结点。

从根开始 BFS 处理 Fail，每个儿子的 Fail 都由当前节点的 Fail 继续跳 Fail 直至对应儿子存在。

AC 自动机

解决多个模板串（Trie）的字符串匹配问题。

Fail 数组

Fail_i 表示 Trie 上 i 点最长的、在 Trie 上出现过的真后缀对应的结点。

从根开始 BFS 处理 Fail，每个儿子的 Fail 都由当前节点的 Fail 继续跳 Fail 直至对应儿子存在。

实际上，实现时直接对每个结点添加不存在的儿子为其 Fail 上该字符的儿子。

时间复杂度 $O(n|\Sigma|)$。

AC 自动机

解决多个模板串（Trie）的字符串匹配问题。

Fail 数组

Fail_i 表示 Trie 上 i 点最长的、在 Trie 上出现过的真后缀对应的结点。

从根开始 BFS 处理 Fail，每个儿子的 Fail 都由当前节点的 Fail 继续跳 Fail 直至对应儿子存在。

实际上，实现时直接对每个结点添加不存在的儿子为其 Fail 上该字符的儿子。

时间复杂度 $O(n|\Sigma|)$。

匹配

用匹配串在 Trie 上从根结点开始走，每次直接走向该字符的后继。在某个位置匹配上的个数为该结点 Fail 树到根的和，可以预处理。

时间复杂度 $O(m)$。

[BZOJ 3172] 单词

给定 n 个字符串，求每个字符串在所有字符串中作为子串共出现了多少次。

$n \leq 200$，字符串总长不超过 10^6，字符集为小写字母。

[BZOJ 3172] 单词

给定 n 个字符串，求每个字符串在所有字符串中作为子串共出现了多少次。

$n \leq 200$，字符串总长不超过 10^6，字符集为小写字母。

解

先对所有串建立 AC 自动机，然后再用每个串在 AC 自动机上匹配。匹配时对 Fail 树到根都有贡献，可以先打一个 tag，最后 Fail 子树求和。

时间复杂度 $O(L)$，其中 L 表示字符串总长。

[BZOJ 2553] 禁忌

给定 n 个禁忌串，随机一个长为 l、字符集为前 a 个字母的字符串，求将其划分为若干段后，禁忌串段数最大值的期望。

$n \le 5$，$l \le 10^9$，$1 \le a \le 26$，每个禁忌串长度不超过 15。

[BZOJ 2553] 禁忌

给定 n 个禁忌串，随机一个长为 l、字符集为前 a 个字母的字符串，求将其划分为若干段后，禁忌串段数最大值的期望。

$n \le 5$，$l \le 10^9$，$1 \le a \le 26$，每个禁忌串长度不超过 15。

解

显然可以贪心匹配，每次匹配完一个禁忌串后将之前的剩余清零即可。

设 $f_{i,j}$ 表示，之后还要 i 个字符、之前字符在 AC 自动机上匹配到状态 j 的匹配数期望。枚举当前字符从 f_{i-1} 转移即可。需要用矩阵乘法优化。

时间复杂度 $O\left((\sum \mathrm{len}_i)^3 \log l\right)$。

[HDU 6096] String

给定 n 个字符串，另有 q 组询问，每次给定两个字符串 p 和 s，求 n 个字符串中有多少个字符串满足以 p 开头、以 s 结尾且 p 和 s 不重叠。

不超过 5 组数据，$n, q \leq 10^5$，给定字符串与询问字符串总长分别不超过 5×10^5，字符集为小写字母。

[HDU 6096] String

给定 n 个字符串，另有 q 组询问，每次给定两个字符串 p 和 s，求 n 个字符串中有多少个字符串满足以 p 开头、以 s 结尾且 p 和 s 不重叠。

不超过 5 组数据，$n, q \leq 10^5$，给定字符串与询问字符串总长分别不超过 5×10^5，字符集为小写字母。

解

将原字符串 T 改为 $\overline{T\#T}$，将询问 p、s 改为字符串 $\overline{s\#p}$，则限制变为带长度限制的匹配。

将字符串与询问分别按照长度排序，用树状数组维护 Fail 树上个数的和，用 AC 自动机匹配即可。

时间复杂度 $O(l \log l)$，其中 l 表示字符串总长。

[CodeForces 547E] Mike and Friends

给定 n 个字符串 $s_1, s_2, \ldots, s_n$，设 $\text{call}(i, j)$ 表示 s_j 在 s_i 的出现次数。q 次询问，每次给定 l、r、k，求 $\sum_{i=l}^{r} \text{call}(i, k)$。

$n \le 2 \times 10^5$，$m \le 5 \times 10^5$，$\sum |s_i| \le 2 \times 10^5$，字符集为小写字母。

[CodeForces 547E] Mike and Friends

给定 n 个字符串 $s_1, s_2, \ldots, s_n$，设 $\mathrm{call}(i, j)$ 表示 s_j 在 s_i 的出现次数。q 次询问，每次给定 l、r、k，求 $\sum_{i=l}^{r} \mathrm{call}(i, k)$。

$n \le 2 \times 10^5$，$m \le 5 \times 10^5$，$\sum |s_i| \le 2 \times 10^5$，字符集为小写字母。

解

在 AC 自动机中，Trie 树的父亲对应前缀，Fail 对应后缀，子串即为前缀的后缀。

[CodeForces 547E] Mike and Friends

给定 n 个字符串 $s_1, s_2, \ldots, s_n$，设 $\operatorname{call}(i, j)$ 表示 s_j 在 s_i 的出现次数。q 次询问，每次给定 l、r、k，求 $\sum_{i=l}^{r} \operatorname{call}(i, k)$。

$n \le 2 \times 10^5$，$m \le 5 \times 10^5$，$\sum |s_i| \le 2 \times 10^5$，字符集为小写字母。

解

在 AC 自动机中，Trie 树的父亲对应前缀，Fail 对应后缀，子串即为前缀的后缀。

因此答案即为 k 对应结点的 Fail 子树中所有 Trie 子树内的 $[l, r]$ 中数的出现次数。

[CodeForces 547E] Mike and Friends

给定 n 个字符串 $s_1, s_2, \ldots, s_n$，设 $\operatorname{call}(i, j)$ 表示 s_j 在 s_i 的出现次数。q 次询问，每次给定 l、r、k，求 $\sum_{i=l}^{r} \operatorname{call}(i, k)$。

$n \le 2 \times 10^5$，$m \le 5 \times 10^5$，$\sum |s_i| \le 2 \times 10^5$，字符集为小写字母。

解

在 AC 自动机中，Trie 树的父亲对应前缀，Fail 对应后缀，子串即为前缀的后缀。

因此答案即为 k 对应结点的 Fail 子树中所有 Trie 子树内的 $[l, r]$ 中数的出现次数。

将询问离线并差分，从小到大枚举 r，一路修改其所有父亲结点的值，在 Fail 树中用 DFS 序和树状数组维护即可。

时间复杂度 $O((\sum |s_i| + m) \log n)$

[BZOJ 1039] 无序运动 Movement

给一个长为 n 的点列，另有 m 个轨迹（点列片段）。对于点列的一个区间，一个轨迹在点列该位置出现，当且仅当这个轨迹通过以下操作可以与该区间**对应位置**完全重合：

平移 选择 d_x、d_y，所有 $(x, y) \to (x + d_x, y + d_y)$；

旋转 选择 t，所有 $(x, y) \to (x \cos t - y \sin t, x \sin t + y \cos t)$；

翻转 所有 $(x, y) \to (x, -y)$；

缩放 选择 $p \neq 0$，所有点 $(x, y) \to (px, py)$。

设 k 表示轨迹长度，$n, k \leq 2 \times 10^5$，$\sum k \leq 1.6 \times 10^6$，坐标为整数且绝对值不超过 10000。

[BZOJ 1039] 无序运动 Movement

给一个长为 n 的点列，另有 m 个轨迹（点列片段）。对于点列的一个区间，一个轨迹在点列该位置出现，当且仅当这个轨迹通过以下操作可以与该区间**对应位置**完全重合：平移、旋转、翻转、缩放。

设 k 表示轨迹长度，$n, k \leq 2 \times 10^5$，$\sum k \leq 1.6 \times 10^6$，坐标为整数且绝对值不超过 10000。

[BZOJ 1039] 无序运动 Movement

给一个长为 n 的点列，另有 m 个轨迹（点列片段）。对于点列的一个区间，一个轨迹在点列该位置出现，当且仅当这个轨迹通过以下操作可以与该区间**对应位置**完全重合：

设 k 表示轨迹长度，$n, k \leq 2 \times 10^5$，$\sum k \leq 1.6 \times 10^6$，坐标为整数且绝对值不超过 10000。

解

将相邻点连线，轨迹可以与点列区间匹配，当且仅当它们相似。

[BZOJ 1039] 无序运动 Movement

给一个长为 n 的点列，另有 m 个轨迹（点列片段）。对于点列的一个区间，一个轨迹在点列该位置出现，当且仅当这个轨迹通过以下操作可以与该区间**对应位置**完全重合：

设 k 表示轨迹长度，$n, k \leq 2 \times 10^5$，$\sum k \leq 1.6 \times 10^6$，坐标为整数且绝对值不超过 10000。

解

将相邻点连线，轨迹可以与点列区间匹配，当且仅当它们相似。

对于每个点记录两条线段的夹角、长度比例，用 AC 自动机匹配即可。

[BZOJ 1039] 无序运动 Movement

给一个长为 n 的点列，另有 m 个轨迹（点列片段）。对于点列的一个区间，一个轨迹在点列该位置出现，当且仅当这个轨迹通过以下操作可以与该区间**对应位置**完全重合：

设 k 表示轨迹长度，$n, k \leq 2 \times 10^5$，$\sum k \leq 1.6 \times 10^6$，坐标为整数且绝对值不超过 10000。

解

将相邻点连线，轨迹可以与点列区间匹配，当且仅当它们相似。

对于每个点记录两条线段的夹角、长度比例，用 AC 自动机匹配即可。

由于字符集较大，要直接暴力跳 Fail 匹配，时间复杂度 $O(n + \sum k)$。

Manacher 算法

给定一个字符串，求出其中的所有极长回文子串。

Manacher 算法

给定一个字符串，求出其中的所有极长回文子串。

解

由于回文串有奇偶两种，我们在字符串中插入 $''\#''$ 分隔，这样所有回文串都变成了奇回文串。

设 l_i 表示以 i 为中心的极长回文半径，再记录 p 使得在已经求出的范围内 $p+l_p$ 最大。

对于当前要处理到的位置 i，如果 $i<p+l_p$，则表示 i 周围的位置与 p 之前的某段对称，可以直接继承那部分的 l。之后再考虑超出 $p+l_p$ 的部分，暴力判断扩充即可。

由于每个位置只会被扩充一次，时间复杂度 $O(n)$。

[Luogu P4555] 最长双回文串

给定一个字符串 S，求其的一个最长子串，满足可以表示成两个回文串的和。

$|S| \leq 10^5$。

[Luogu P4555] 最长双回文串

给定一个字符串 S，求其的一个最长子串，满足可以表示成两个回文串的和。

$|S| \leq 10^5$。

解

设 l_i、r_i 分别表示以 i 为左、右端点的最长回文串，那么这个要么是某个极长回文串的端点，要么可以又上一位递推得到。

极长回文串可以用 Manacher 算法求出，然后枚举中间位置统计答案即可。

时间复杂度 $O(|S|)$。

[HDU 6230] Palindrome

一个字符串 $s_{1\ldots 3n-2}$ 被称为 one-and-half palindromic，当且仅当其满足 $s_i = s_{2n-i} = s_{2n+i-2}$ $(\forall 1 \leq i \leq n)$。给定一个字符串 T，求其有多少个子串是 one-and half palindromic。

$|T| \leq 5 \times 10^5$，字符集为小写字母。

[HDU 6230] Palindrome

一个字符串 $s_{1\ldots 3n-2}$ 被称为 one-and-half palindromic，当且仅当其满足 $s_i = s_{2n-i} = s_{2n+i-2}$ $(\forall 1 \le i \le n)$。给定一个字符串 T，求其有多少个子串是 one-and half palindromic。

$|T| \le 5 \times 10^5$，字符集为小写字母。

解

one-and-half palindromic 串一定形如 $a\overrightarrow{S}b\overleftarrow{S}a\overrightarrow{S}b$。

[HDU 6230] Palindrome

一个字符串 $s_{1\ldots 3n-2}$ 被称为 one-and-half palindromic，当且仅当其满足 $s_i = s_{2n-i} = s_{2n+i-2}$ $(\forall 1 \le i \le n)$。给定一个字符串 T，求其有多少个子串是 one-and half palindromic。

$|T| \le 5 \times 10^5$，字符集为小写字母。

解

one-and-half palindromic 串一定形如 $a\overrightarrow{S}b\overleftarrow{S}a\overrightarrow{S}b$。

考虑两个回文中心 $i < j$，则要求 $j - p_j + 1 \le i < j \le i + p_i - 1$。

枚举 j，则要求 $i \in [j - p_j + 1, j - 1]$ 且 $i + p_i - 1 \ge j$。

[HDU 6230] Palindrome

一个字符串 $s_{1\dots3n-2}$ 被称为 one-and-half palindromic，当且仅当其满足 $s_i = s_{2n-i} = s_{2n+i-2}$ $(\forall 1 \le i \le n)$。给定一个字符串 T，求其有多少个子串是 one-and half palindromic。

$|T| \le 5 \times 10^5$，字符集为小写字母。

解

one-and-half palindromic 串一定形如 $a\overrightarrow{S}b\overleftarrow{S}a\overrightarrow{S}b$。

考虑两个回文中心 $i < j$，则要求 $j - p_j + 1 \le i < j \le i + p_i - 1$。

枚举 j，则要求 $i \in [j - p_j + 1, j - 1]$ 且 $i + p_i - 1 \ge j$。

从大到小枚举 j、再将 i 按照 $i + p_i - 1$ 排序插入，i 的区间限制可以用树状数组维护。

时间复杂度 $O(|T| \log |T|)$。

扩展 KMP（Z 算法）

给定字符串 S，设 Z_i 表示 S 和 $S_{i...n-1}$ 的最长公共前缀，求 Z。

扩展 KMP（Z 算法）

给定字符串 S，设 Z_i 表示 S 和 $S_{i\dots n-1}$ 的最长公共前缀，求 Z。

解

与 Manacher 类似地，我们记录当前 $p+Z_p$ 最大的 p。

当计算到 i 时，如果 $i<p+Z_p$，那么说明 i 之后这一段与之前一段相同，可以直接继承该处的 Z。之后再考虑超出 $p+Z_p$ 的部分，暴力扩充即可。

时间复杂度 $O(n)$。

扩展 KMP（Z 算法）

给定字符串 S，设 Z_i 表示 S 和 $S_{i...n-1}$ 的最长公共前缀，求 Z。

解

与 Manacher 类似地，我们记录当前 $p+Z_p$ 最大的 p。

当计算到 i 时，如果 $i < p + Z_p$，那么说明 i 之后这一段与之前一段相同，可以直接继承该处的 Z。之后再考虑超出 $p+Z_p$ 的部分，暴力扩充即可。

时间复杂度 $O(n)$。

匹配

给定字符串 T，分别求 T 与 S 所有后缀的最长公共前缀。

扩展 KMP（Z 算法）

给定字符串 S，设 Z_i 表示 S 和 $S_{i \dots n-1}$ 的最长公共前缀，求 Z。

解

与 Manacher 类似地，我们记录当前 $p + Z_p$ 最大的 p。

当计算到 i 时，如果 $i < p + Z_p$，那么说明 i 之后这一段与之前一段相同，可以直接继承该处的 Z。之后再考虑超出 $p + Z_p$ 的部分，暴力扩充即可。

时间复杂度 $O(n)$。

匹配

给定字符串 T，分别求 T 与 S 所有后缀的最长公共前缀。

方法与上面类似，仍然使用 Z 来继承，继承之前位置与 T 的最长公共前缀即可。

[CodeForces 149E] Martian Strings

给定一个长为 n 的字符串 S，另给定 m 个模式串，求其中有多少个模式串 T 满足 $T = S_{a...b} + S_{c...d}$（$a \leq b < c \leq d$）。

$n \leq 10^5$，$m \leq 100$，每个模式串长度不超过 1000，字符集为大写字母。

[CodeForces 149E] Martian Strings

给定一个长为 n 的字符串 S，另给定 m 个模式串，求其中有多少个模式串 T 满足 $T = S_{a...b} + S_{c...d}$ ($a \le b < c \le d$)。

$n \le 10^5$，$m \le 100$，每个模式串长度不超过 1000，字符集为大写字母。

解

先枚举每个模式串，然后枚举 a，显然贪心地匹配尽量大的 b 是最优的，这个长度可以通过扩展 KMP 求出。接下来需要判断 b 之后的部分是否存在位置能匹配其余部分，可以将串反转后同样地扩展 KMP 解决。

时间复杂度 $O(nm)$。

[CodeForces 1051E] Vasya and Big Integers

给一个长为 n 的数字串 S（不含前导 0），求有多少种方案把 S 划分成若干个没有前导 0 的数，使得每部分都在 $[l, r]$ 中。答案对 998244353 取模的结果。

$n \le 10^6$，$0 \le l \le r \le 10^{1000000}$。

[CodeForces 1051E] Vasya and Big Integers

给一个长为 n 的数字串 S（不含前导 0），求有多少种方案把 S 划分成若干个没有前导 0 的数，使得每部分都在 $[l, r]$ 中。答案对 998244353 取模的结果。

$n \le 10^6$，$0 \le l \le r \le 10^{1000000}$。

解

DP，设 f_i 表示前 i 位的划分方案数。枚举最后一段划分 $S_{j\dots i}$ 满足条件，用 f_j 转移即可。

朴素实现时间复杂度不低于 $O(n^2)$。

[CodeForces 1051E] Vasya and Big Integers

给一个长为 n 的数字串 S（不含前导 0），求有多少种方案把 S 划分成若干个没有前导 0 的数，使得每部分都在 $[l, r]$ 中。答案对 998244353 取模的结果。

$n \leq 10^6$，$0 \leq l \leq r \leq 10^{1000000}$。

解

DP，设 f_i 表示前 i 位的划分方案数。枚举最后一段划分 $S_{j...i}$ 满足条件，用 f_j 转移即可。

朴素实现时间复杂度不低于 $O(n^2)$。

事实上，对于长度在 l、r 之间的长度，一定可以转移，这个区间求和问题用前缀和优化即可；对于 l、r 的位置，可以用扩展 KMP 求出 LCP 比较大小判断。

时间复杂度 $O(n)$。