AC 自动机，Trie 图

问题背景

背景

给出一个字典，和若干询问：多少个字典串在询问串中出现过。
即单串与多串的匹配问题。

暴力算法

KMP 的打法

将问题看作是：每个字典串与询问串的单模式匹配。

暴力算法

KMP 的打法

将问题看作是：每个字典串与询问串的单模式匹配。
缺点：字典串形成了孤岛，没有打通链路，形成闭环，导致复杂度很高。

暴力算法

KMP 的打法

将问题看作是：每个字典串与询问串的单模式匹配。
缺点：字典串形成了孤岛，没有打通链路，形成闭环，导致复杂度很高。

Trie 的打法

利用字典树进行多模式匹配。

暴力算法

KMP 的打法

将问题看作是：每个字典串与询问串的单模式匹配。
缺点：字典串形成了孤岛，没有打通链路，形成闭环，导致复杂度很高。

Trie 的打法

利用字典树进行多模式匹配。
缺点：不能够在失配时进行合理的跳转，盲目尝试，导致复杂度很高。。

AC 自动机

AC 自动机 = Trie + KMP

AC 自动机基于 Trie，将 KMP 的 Border 概念推广到多模式串上。

AC 自动机

AC 自动机 = Trie + KMP

AC 自动机基于 Trie，将 KMP 的 Border 概念推广到多模式串上。
AC 自动机是一种离线型数据结构，即不支持增量添加新的字符串。

AC 自动机

AC 自动机 = Trie + KMP

AC 自动机基于 Trie，将 KMP 的 Border 概念推广到多模式串上。
AC 自动机是一种离线型数据结构，即不支持增量添加新的字符串。
AC 自动机常用于将字符串询问类的问题进行离线处理，也经常与各种 DP 结合，或是补全成 Trie 图。

border 概念的推广

广义 border

推广到两个串：对于两个串 S 和 T，相等的 p 长度的 S 的后缀和 T 的前缀称为一个 border。

border 概念的推广

广义 border

推广到两个串：对于两个串 S 和 T，相等的 p 长度的 S 的后缀和 T 的前缀称为一个 border。

推广到一个字典：对于串 S 和一个字典 D，相等的 p 长度的 S 的后缀，和任意一个字典串 T 的前缀称为一个 border。

border 概念的推广

广义 border

推广到两个串：对于两个串 S 和 T，相等的 p 长度的 S 的后缀和 T 的前缀称为一个 border。

推广到一个字典：对于串 S 和一个字典 D，相等的 p 长度的 S 的后缀，和任意一个字典串 T 的前缀称为一个 border。

失配（Fail）指针：对于 Trie 中的每一个节点（即某个字典串的前缀），它与 Trie 中所有串的最大 Border 即为失配指针。

AC 自动机

失配指针

类似与 KMP 求 Border，任意节点的 Border 长度减一，一定是父节点的 Border。因此可以通过遍历父节点的失配指针链来求解。

AC 自动机

失配指针

类似与 KMP 求 Border，任意节点的 Border 长度减一，一定是父节点的 Border。因此可以通过遍历父节点的失配指针链来求解。
因此在求失配指针的时候，一定要按长度从小到大来求，即 bfs。

AC 自动机

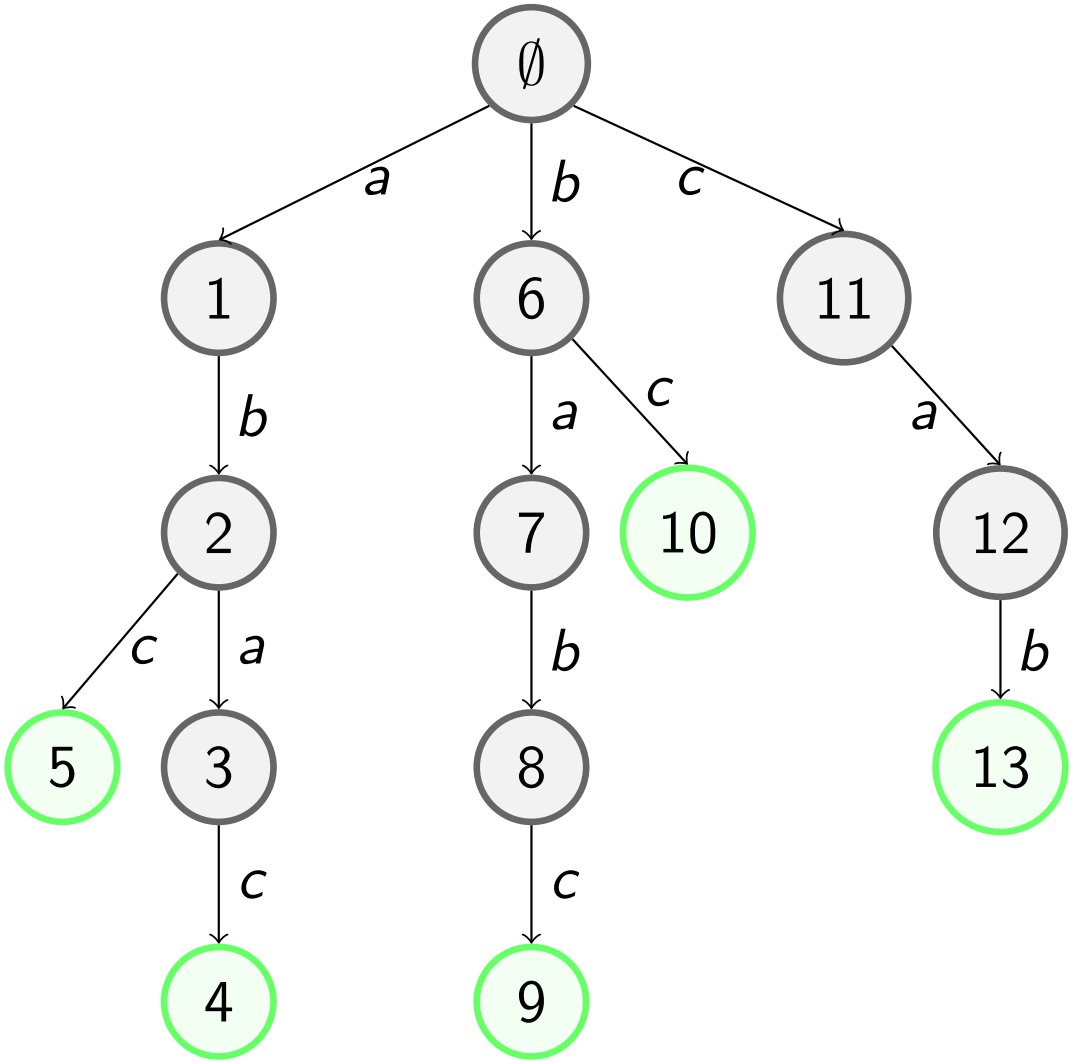
失配指针

类似与 KMP 求 Border，任意节点的 Border 长度减一，一定是父节点的 Border。因此可以通过遍历父节点的失配指针链来求解。
因此在求失配指针的时候，一定要按长度从小到大来求，即 bfs。

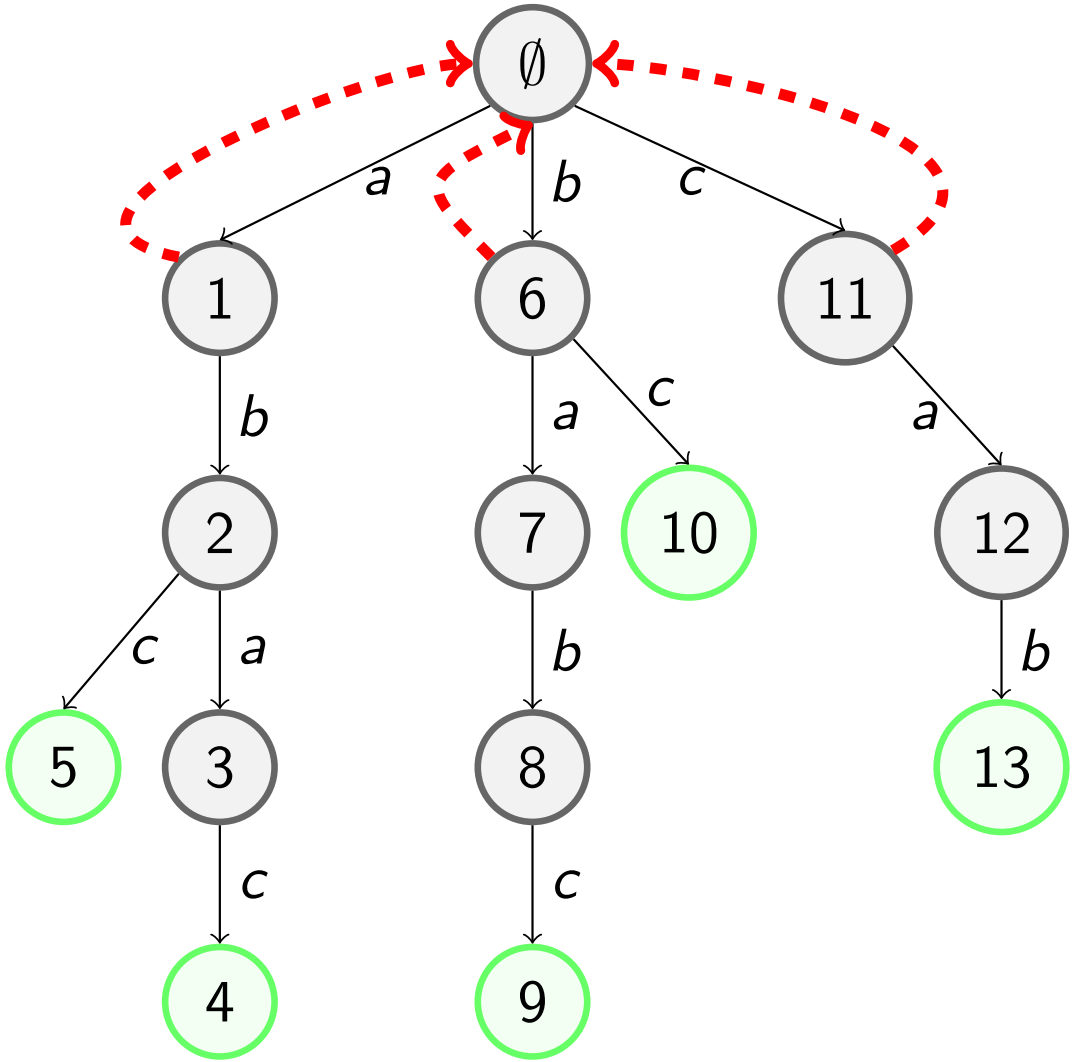
复杂度分析

类似于 KMP 的势能分析方法，势能总量等于 Trie 的节点总数，因此复杂度为线性的。

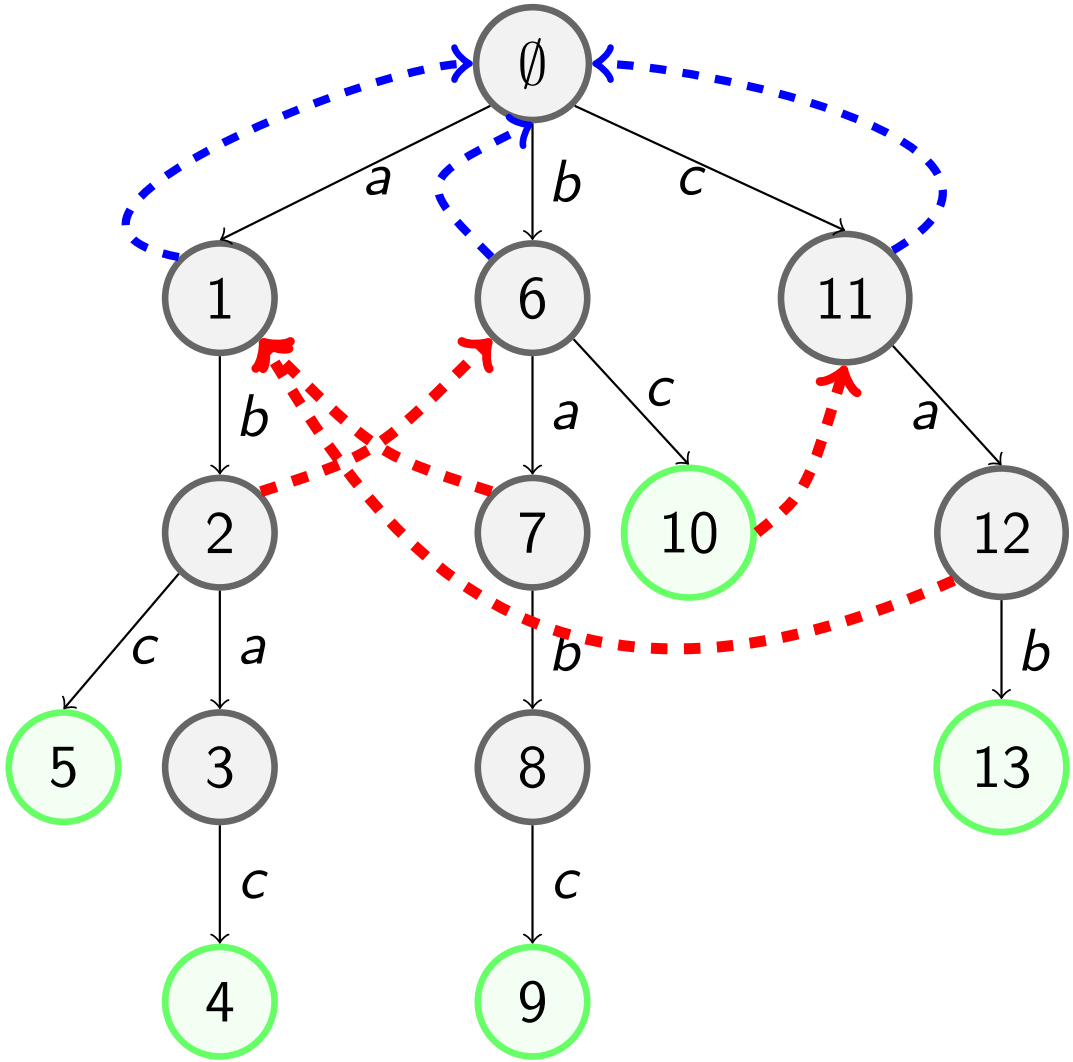
Trie



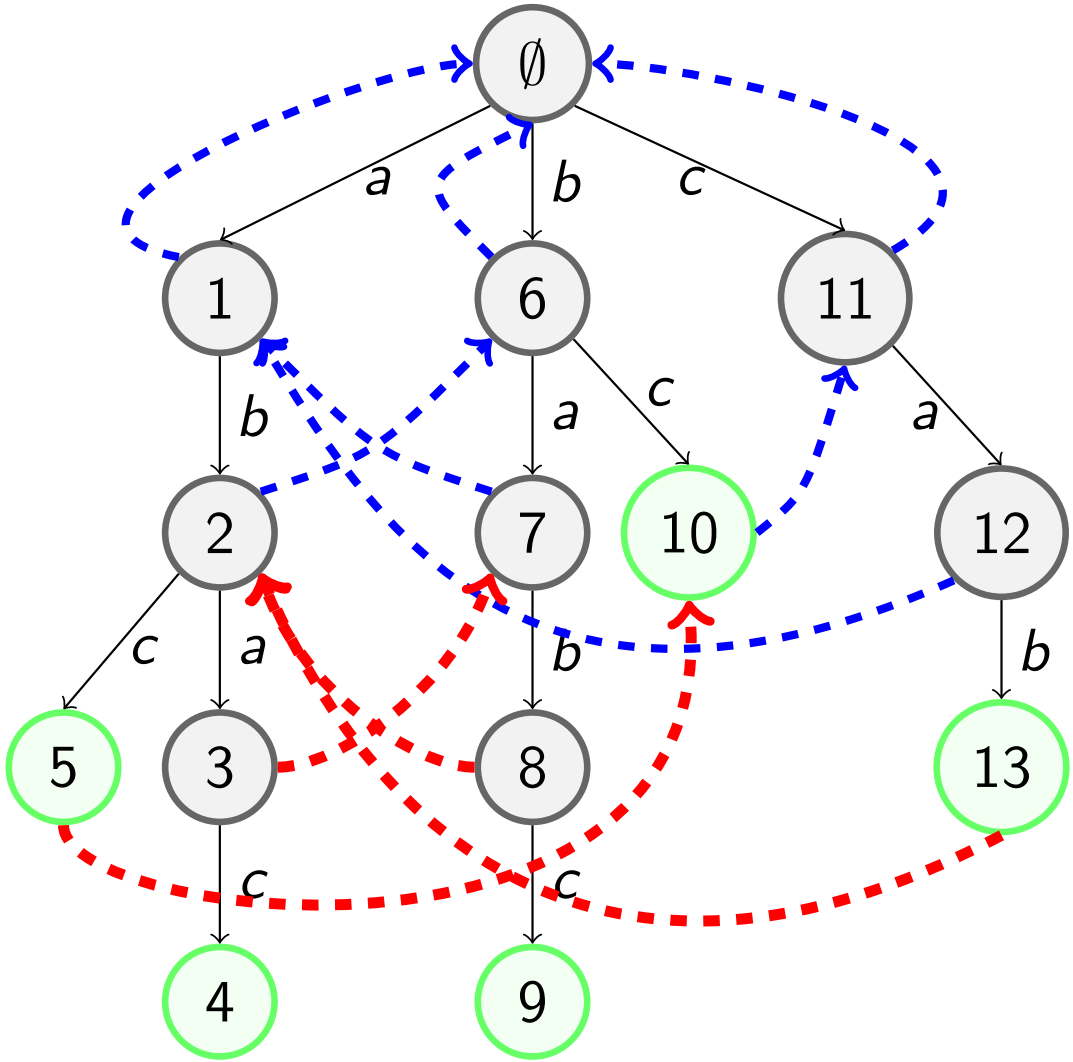
求失配指针-Layer 1



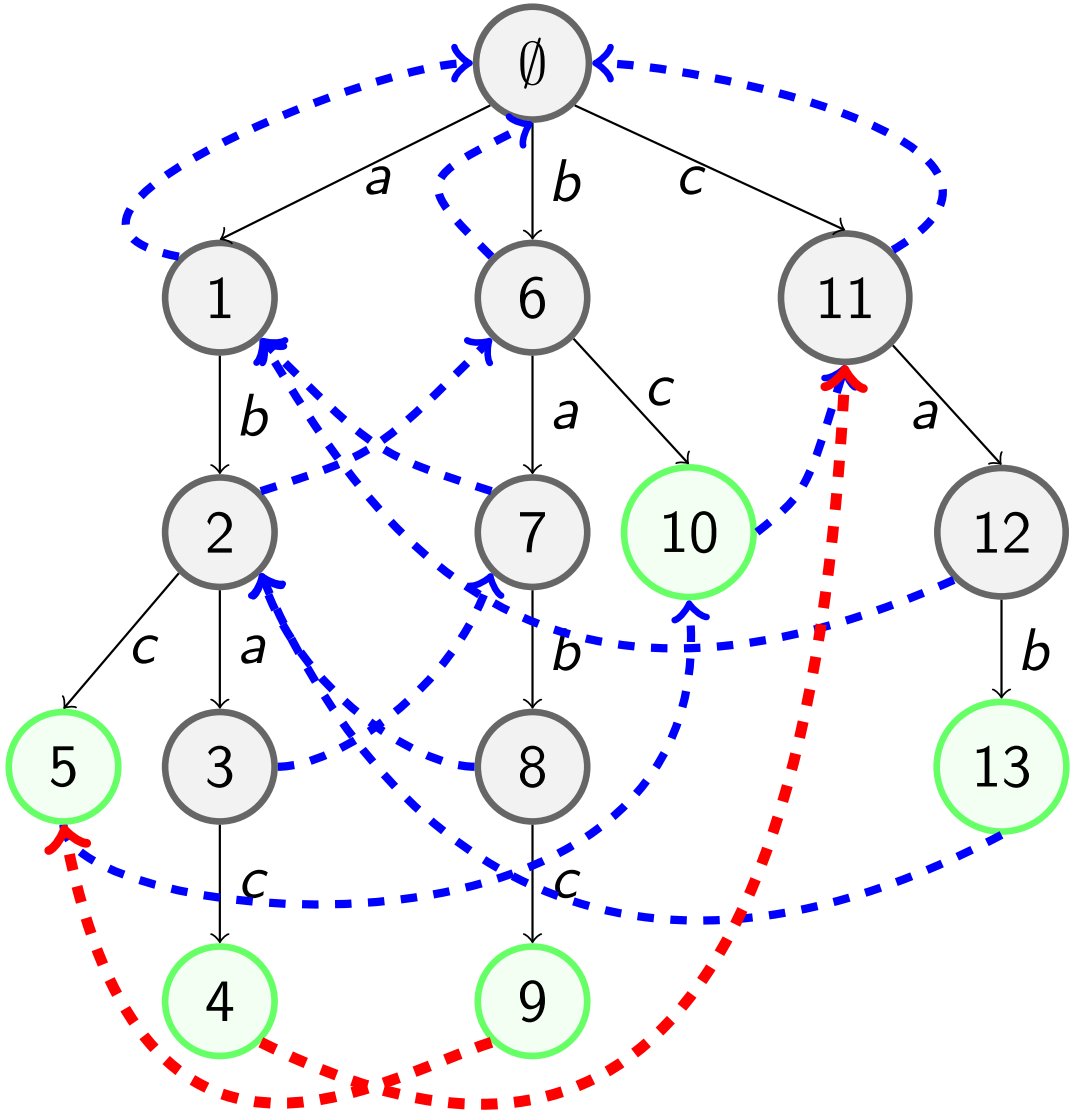
求失配指针-Layer 2



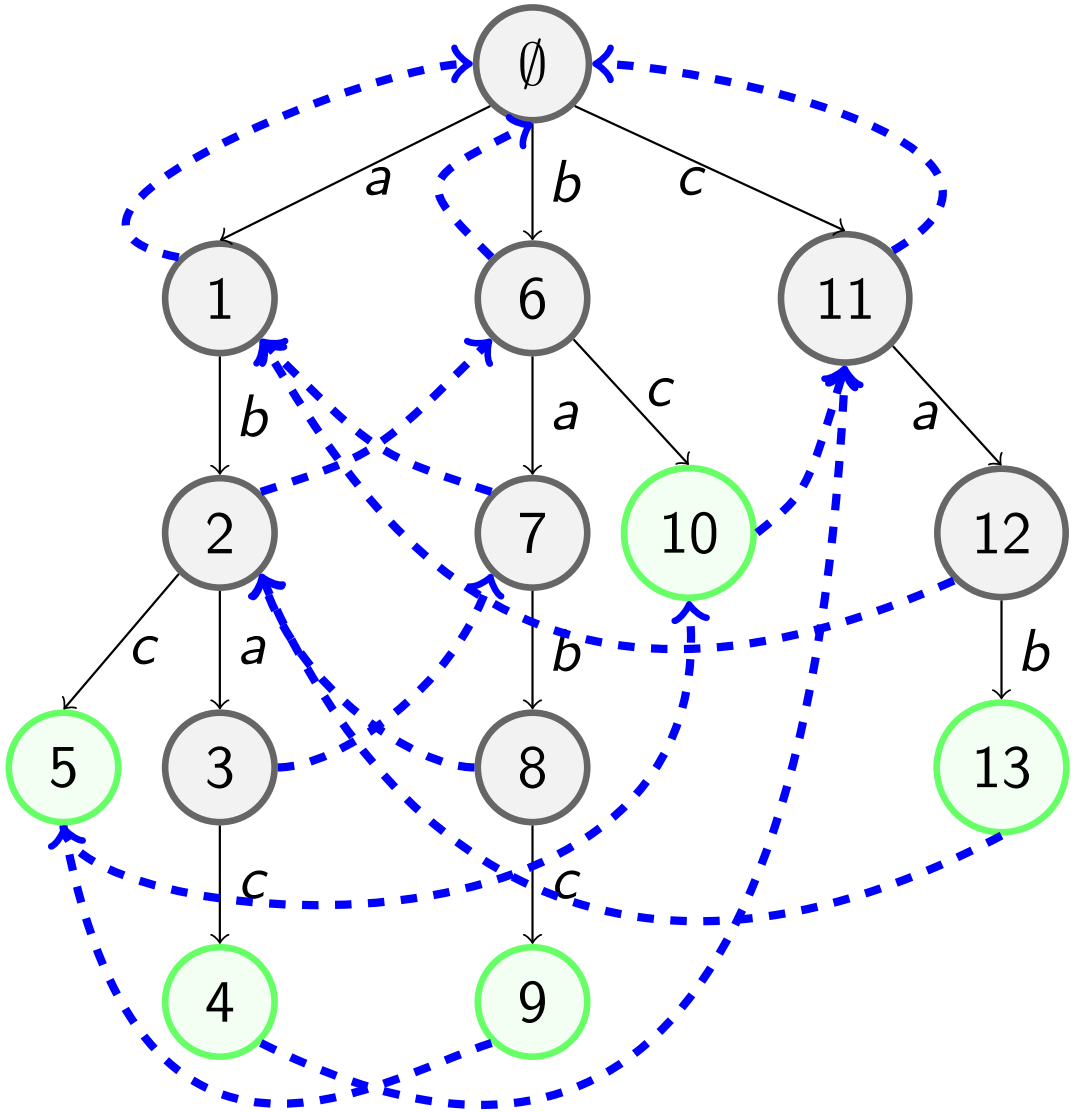
求失配指针-Layer 3



求失配指针-Layer 4



AC 自动机



例题 1

某谷 P5357

给出一个文本串 S 和一个字典，求每个字典串在 S 中的出现次数。

例题 1

某谷 P5357

给出一个文本串 S 和一个字典，求每个字典串在 S 中的出现次数。

题解

将文本串放在 AC 自动机上运行，求出每个前缀匹配到 AC 自动机的哪个节点，将该节点的标记值 $+1$。

例题 1

某谷 P5357

给出一个文本串 S 和一个字典，求每个字典串在 S 中的出现次数。

题解

将文本串放在 AC 自动机上运行，求出每个前缀匹配到 AC 自动机的哪个节点，将该节点的标记值 $+1$。
每个字典串的出现次数等于失配树的子树内的标记总量。因此在失配树上，自底向上推标记即可。

例题 2

牛客 14612

给出一个有 N 个串的字典，然后进行 M 次操作：
1. 修改操作：向字典中新增一个字典串 S。
2. 查询操作：查询所有字典串在询问串 T 中的出现次数，同一个字典串重复出现算多次。

$1 \leq N, M \leq 100,000$，输入字符串总长度不超过 $3000,000$。

例题 2

牛客 14612

给出一个有 N 个串的字典，然后进行 M 次操作：
1. 修改操作：向字典中新增一个字典串 S。
2. 查询操作：查询所有字典串在询问串 T 中的出现次数，同一个字典串重复出现算多次。

$1 \leq N, M \leq 100,000$，输入字符串总长度不超过 $3000,000$。

题解

先无视修改操作，只考虑查询操作：可以建出 AC 自动机，然后将查询串在 AC 自动机上运行，匹配到 AC 自动机的一个节点时，将该节点到根的路径上终止标记个数计入答案。

例题 2

牛客 14612

给出一个有 N 个串的字典，然后进行 M 次操作：
1. 修改操作：向字典中新增一个字典串 S。
2. 查询操作：查询所有字典串在询问串 T 中的出现次数，同一个字典串重复出现算多次。

$1 \leq N, M \leq 100,000$，输入字符串总长度不超过 $3000,000$。

题解

先无视修改操作，只考虑查询操作：可以建出 AC 自动机，然后将查询串在 AC 自动机上运行，匹配到 AC 自动机的一个节点时，将该节点到根的路径上终止标记个数计入答案。

由于 AC 自动机是**离线型数据结构**，即不能支持快速增删字典串，因此考虑到修改操作需要先离线，动态维护终止标记，使用 dfs 序 + 树状数组快速维护每个点到根路径上的终止标记个数。

例题 3

某谷 P7456

给出一个字典，至多包含 $300,000$ 个字典串，且字典串总长度不超过 $300,000$。

再给出一个文本串 $T(|T| \leq 300,000)$，问最少使用多少个字典串可以拼接出 T。同一个字典串使用多次算多次。

拼接时允许重叠，只需要保证重叠部分匹配即可，例如 $T = abcd$，可以使用字典串 $S = cdxx$ 拼接成 $T' = abcdxx$。

例题 3

某谷 P7456

给出一个字典，至多包含 $300,000$ 个字典串，且字典串总长度不超过 $300,000$。
再给出一个文本串 $T(|T| \leq 300,000)$，问最少使用多少个字典串可以拼接出 T。同一个字典串使用多次算多次。
拼接时允许重叠，只需要保证重叠部分匹配即可，例如 $T = abcd$，可以使用字典串 $S = cdxx$ 拼接成 $T' = abcdxx$。

题解

首先比较容易想到使用 DP：$f[i]$ 表示拼接出 $T[1,i]$ 的操作次数.

例题 3

某谷 P7456

给出一个字典，至多包含 $300,000$ 个字典串，且字典串总长度不超过 $300,000$。
再给出一个文本串 $T(|T| \leq 300,000)$，问最少使用多少个字典串可以拼接出 T。同一个字典串使用多次算多次。
拼接时允许重叠，只需要保证重叠部分匹配即可，例如 $T = abcd$，可以使用字典串 $S = cdxx$ 拼接成 $T' = abcdxx$。

题解

首先比较容易想到使用 DP：$f[i]$ 表示拼接出 $T[1,i]$ 的操作次数.
转移时，设某字典串 S_i 等于 $T[1,i]$ 的后缀，会导致如下转移：

例题 3

某谷 P7456

给出一个字典，至多包含 $300,000$ 个字典串，且字典串总长度不超过 $300,000$。
再给出一个文本串 $T(|T| \leq 300,000)$，问最少使用多少个字典串可以拼接出 T。同一个字典串使用多次算多次。
拼接时允许重叠，只需要保证重叠部分匹配即可，例如 $T = abcd$，可以使用字典串 $S = cdxx$ 拼接成 $T' = abcdxx$。

题解

首先比较容易想到使用 DP：$f[i]$ 表示拼接出 $T[1,i]$ 的操作次数.
转移时，设某字典串 S_i 等于 $T[1,i]$ 的后缀，会导致如下转移：
$f[i] = \min(f[i-1], f[i-2], \cdots, f[i-|S_i|]) + 1$。因此只需要寻找长度最长的字典串即可，使用 AC 自动机可以解决。

例题 3

某谷 P7456

给出一个字典，至多包含 $300,000$ 个字典串，且字典串总长度不超过 $300,000$。
再给出一个文本串 $T(|T| \leq 300,000)$，问最少使用多少个字典串可以拼接出 T。同一个字典串使用多次算多次。
拼接时允许重叠，只需要保证重叠部分匹配即可，例如 $T = abcd$，可以使用字典串 $S = cdxx$ 拼接成 $T' = abcdxx$。

题解

首先比较容易想到使用 DP：$f[i]$ 表示拼接出 $T[1,i]$ 的操作次数.
转移时，设某字典串 S_i 等于 $T[1,i]$ 的后缀，会导致如下转移：
$f[i] = \min(f[i-1], f[i-2], \cdots, f[i-|S_i|]) + 1$。因此只需要寻找长度最长的字典串即可，使用 AC 自动机可以解决。
然后使用线段树辅助进行 DP 转移。
复杂度 $O(300000 \cdot (26 + \log n))$。

例题 4

牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 100) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

例题 4

牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 100) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

题解

答案 $= 26^M$ - 完全不包含字典串的串个数。

例题 4

牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 100) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

题解

答案 $= 26^M$ - 完全不包含字典串的串个数。
包含与不包含本质上都是字符串匹配问题，因此本题为字符串多模式匹配，很容易想到利用 AC 自动机。

例题 4

牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 100) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

题解

答案 $= 26^M$ - 完全不包含字典串的串个数。
包含与不包含本质上都是字符串匹配问题，因此本题为字符串多模式匹配，很容易想到利用 AC 自动机。
设 $f[len][u]$ 表示长度为 len 的字符串，当前匹配到了 AC 自动机的 u 节点的方案数。

例题 4

牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 100) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

题解

答案 $= 26^M$ - 完全不包含字典串的串个数。
包含与不包含本质上都是字符串匹配问题，因此本题为字符串多模式匹配，很容易想到利用 AC 自动机。
设 $f[len][u]$ 表示长度为 len 的字符串，当前匹配到了 AC 自动机的 u 节点的方案数。
转移时枚举下一个字符，然后找到匹配的 AC 自动机节点 v(这一步的复杂度为 $O(depth)$)，如果 v 是一个终止节点，则是一个不合法的转移。

例题 4

牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 100) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

题解

答案 $= 26^M$ - 完全不包含字典串的串个数。
包含与不包含本质上都是字符串匹配问题，因此本题为字符串多模式匹配，很容易想到利用 AC 自动机。
设 $f[len][u]$ 表示长度为 len 的字符串，当前匹配到了 AC 自动机的 u 节点的方案数。
转移时枚举下一个字符，然后找到匹配的 AC 自动机节点 v(这一步的复杂度为 $O(depth)$)，如果 v 是一个终止节点，则是一个不合法的转移。
复杂度 $O(M \cdot 100 \cdot 100 \cdot 26) = 2.7 \cdot 10^7$

例题 5-Trie 图

加强版牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 10000) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

例题 5-Trie 图

加强版牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 10000) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

Trie 图

用上题的做法复杂度高达 $2.7 \cdot 10^9$ 不能通过。因此需要进行优化：

例题 5-Trie 图

加强版牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 10000) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

Trie 图

用上题的做法复杂度高达 $2.7 \cdot 10^9$ 不能通过。因此需要进行优化：
可以比较容易的观察到可以与处理出 $trans[u][ch]$ =AC 自动机节点 u，后面添加字符 ch 后会转移到哪个节点。**如此可以省掉跳 Fail 链的复杂度。**

例题 5-Trie 图

加强版牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 10000) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

Trie 图

用上题的做法复杂度高达 $2.7 \cdot 10^9$ 不能通过。因此需要进行优化：
可以比较容易的观察到可以与处理出 $trans[u][ch]$ =AC 自动机节点 u，后面添加字符 ch 后会转移到哪个节点。**如此可以省掉跳 Fail 链的复杂度。**
求 $trans[u][ch]$ 时，讨论：
1. 如果 u 节点有 ch 后继边，则 $trans[u][ch] = nxt[u][ch]$。
2. 否则，$trans[u][ch] = trans[Fail[u]][ch]$。

例题 5-Trie 图

加强版牛客 20155

给出一个字典，至多包含 60 个字典串，且字典串总长度不超过 100。求所有长度为 M(≤ 10000) 的串中（共 26^M 个），有多少个串至少包含一个字典串。

Trie 图

用上题的做法复杂度高达 $2.7 \cdot 10^9$ 不能通过。因此需要进行优化：
可以比较容易的观察到可以与处理出 $trans[u][ch]$ =AC 自动机节点 u，后面添加字符 ch 后会转移到哪个节点。**如此可以省掉跳 Fail 链的复杂度。**
求 $trans[u][ch]$ 时，讨论：
1. 如果 u 节点有 ch 后继边，则 $trans[u][ch] = nxt[u][ch]$。
2. 否则，$trans[u][ch] = trans[Fail[u]][ch]$。

因此 $trans$ 数组也需要使用 bfs 来求，可以和 AC 自动机的构造放在一起进行。

例题 6-Trie 图

某谷 P2444

给出 $N \leq 2,000$ 个 01 串，它们每一个表示一段病毒序列，病毒代码总长度不超过 $30,000$。
求是否存在无限长度的 01 串，不包含任意病毒序列作为子串。

例题 6-Trie 图

某谷 P2444

给出 $N \leq 2,000$ 个 01 串，它们每一个表示一段病毒序列，病毒代码总长度不超过 $30,000$。
求是否存在无限长度的 01 串，不包含任意病毒序列作为子串。

题解

本题本质是问：Trie 图上（去掉所有终止节点）是否存在无限长度的路径，充要条件为：有环。

例题 6-Trie 图

某谷 P2444

给出 $N \leq 2,000$ 个 01 串，它们每一个表示一段病毒序列，病毒代码总长度不超过 $30,000$。
求是否存在无限长度的 01 串，不包含任意病毒序列作为子串。

题解

本题本质是问：Trie 图上（去掉所有终止节点）是否存在无限长度的路径，充要条件为：有环。
因此本题需要先构造 AC 自动机，然后补全成为 Trie 图，使用拓扑排序判定是否有环。
注意，这个环必须是 *root* 节点可达的。
复杂度 $O(30,000 \cdot 2)$。

例题 7

牛客 20366

我们称一个正整数 N 是幸运数，当且仅当它的十进制表示中不包含数字串集合 S 中任意一个元素作为其子串。例如当 S=$\{22, 333, 0233\}$ 时，233 是幸运数，2333、20233、3223 不是幸运数。
给定 N 和 S，计算不大于 N 的幸运数个数。
$1 \leq n \leq 10^{1201}$，S 中至多有 100 个串，总长度不超过 $1,500$。

例题 7

牛客 20366

我们称一个正整数 N 是幸运数，当且仅当它的十进制表示中不包含数字串集合 S 中任意一个元素作为其子串。例如当 S=$\{22, 333, 0233\}$ 时，233 是幸运数，2333、20233、3223 不是幸运数。
给定 N 和 S，计算不大于 N 的幸运数个数。
$1 \leq n \leq 10^{1201}$，S 中至多有 100 个串，总长度不超过 $1,500$。

题解

本题与上一题的唯一区别是生硬地结合了数位 DP。
令 $f[u][len][all_zero][limit]$ 表示：长度为 len 的数字串，匹配到 AC 自动机的 u 节点，$all_zero = 0/1$ 表示是否数字串全零，$limit = 0/1$ 表示数字串是否等于 N 的前缀的方案数，注意如果 $all_zero = 1$，数字串匹配到的 AC 自动机节点必须是 $root$。
需要将 AC 自动机补全为 Trie 图，复杂度 $O(1201 \cdot 1500 \cdot 2 \cdot 2 \cdot 26) = 1.8 \cdot 10^8$。实际上并没有这么高。

例题 8

CF547E

给出 N 个字符串，设 $f(S, T)$ 表示 T 在 S 中的出现次数。
给出 Q 次询问：
给定 L, R, K，求 $\sum_{i=L}^{i\leq R} f(S_i, S_K)$

例题 8

CF547E

给出 N 个字符串，设 $f(S, T)$ 表示 T 在 S 中的出现次数。
给出 Q 次询问：
给定 L, R, K，求 $\sum_{i=L}^{i \leq R} f(S_i, S_K)$

题解

本题做法很多，其中一种做法是《AC 自动机 Fail 树 Dfs 序上建可持久化线段树》。

例题 8

CF547E

给出 N 个字符串，设 $f(S, T)$ 表示 T 在 S 中的出现次数。
给出 Q 次询问：
给定 L, R, K，求 $\sum_{i=L}^{i \leq R} f(S_i, S_K)$

题解

本题做法很多，其中一种做法是《AC 自动机 Fail 树 Dfs 序上建可持久化线段树》。
先建出 AC 自动机，在回答询问的时候先找到 S_K 所在的节点 u，那么完整包含字符串 S_K 的状态节点一定位于 u 的 Fail 树子树内。

例题 8

CF547E

给出 N 个字符串，设 $f(S, T)$ 表示 T 在 S 中的出现次数。
给出 Q 次询问：
给定 L, R, K，求 $\sum_{i=L}^{i\leq R} f(S_i, S_K)$

题解

本题做法很多，其中一种做法是《AC 自动机 Fail 树 Dfs 序上建可持久化线段树》。
先建出 AC 自动机，在回答询问的时候先找到 S_K 所在的节点 u，那么完整包含字符串 S_K 的状态节点一定位于 u 的 Fail 树子树内。
即问题等价于询问 u 的 Fail 树子树内，$S_L, S_{L+1}, \cdots, S_R$ 的前缀节点个数有多少。

例题 8

CF547E

给出 N 个字符串，设 $f(S,T)$ 表示 T 在 S 中的出现次数。
给出 Q 次询问：
给定 L,R,K，求 $\sum_{i=L}^{i\leq R} f(S_i,S_K)$

题解

本题做法很多，其中一种做法是《AC 自动机 Fail 树 Dfs 序上建可持久化线段树》。
先建出 AC 自动机，在回答询问的时候先找到 S_K 所在的节点 u，那么完整包含字符串 S_K 的状态节点一定位于 u 的 Fail 树子树内。
即问题等价于询问 u 的 Fail 树子树内，$S_L, S_{L+1}, \cdots, S_R$ 的前缀节点个数有多少。
子树求和，比较容易转化为 Dfs 序的区间求和，对字典串的区间询问，容易转化成持久化数据结构。即 Dfs 序上建可持久化线段树。

例题 9

ICPC WF 2019G First of Her Name

给出 n 个人，他们每个人的名字都是之前某个人的名字在最前边加上一个字母得到。比如 2 号的名字是 AC，3 号在 2 号前边加一个字母 K，这样 3 号的名字是 KAC。
之后给出 k 次询问，每次给出一个字符串 T，询问名字前缀是 T 的有几个人。

例题 9

ICPC WF 2019G First of Her Name

给出 n 个人，他们每个人的名字都是之前某个人的名字在最前边加上一个字母得到。比如 2 号的名字是 AC，3 号在 2 号前边加一个字母 K，这样 3 号的名字是 KAC。
之后给出 k 次询问，每次给出一个字符串 T，询问名字前缀是 T 的有几个人。

题解

本题题解大致有三种：广义 SAM，树上 SA，离线 AC 自动机。

例题 9

ICPC WF 2019G First of Her Name

给出 n 个人，他们每个人的名字都是之前某个人的名字在最前边加上一个字母得到。比如 2 号的名字是 AC，3 号在 2 号前边加一个字母 K，这样 3 号的名字是 KAC。
之后给出 k 次询问，每次给出一个字符串 T，询问名字前缀是 T 的有几个人。

题解

本题题解大致有三种：广义 SAM，树上 SA，离线 AC 自动机。
不难看出输入是一个 Trie 树，但是每个点表示的串是反过来的：从这个点到根，而不是从根到这个点。可以简单将所有人名字反过来，同时将询问也反过来。
即询问有多少个 Trie 节点表示的（正）串以询问串的反串 T' 为后缀。

例题 9

题解

在 AC 自动机上，以某个节点 u 表示的串作为后缀的串，全部位于该节点的 Fail 树子树内。因此本题的思路依然是将文本串放在 AC 自动机上运行（匹配），然后自底向上推标记。

例题 9

题解

在 AC 自动机上，以某个节点 u 表示的串作为后缀的串，全部位于该节点的 Fail 树子树内。因此本题的思路依然是将文本串放在 AC 自动机上运行（匹配），然后自底向上推标记。
离线对所有询问串的反串建 AC 自动机。

例题 9

题解

在 AC 自动机上，以某个节点 u 表示的串作为后缀的串，全部位于该节点的 Fail 树子树内。因此本题的思路依然是将文本串放在 AC 自动机上运行（匹配），然后自底向上推标记。
离线对所有询问串的反串建 AC 自动机。
然后将整个 Trie 树放在 AC 自动机上运行：即找到每个 Trie 树节点表示的串匹配到 AC 自动机的哪个节点。可以用 bfs/dfs 遍历 Trie 树。

例题 9

题解

在 AC 自动机上，以某个节点 u 表示的串作为后缀的串，全部位于该节点的 Fail 树子树内。因此本题的思路依然是将文本串放在 AC 自动机上运行（匹配），然后自底向上推标记。
离线对所有询问串的反串建 AC 自动机。
然后将整个 Trie 树放在 AC 自动机上运行：即找到每个 Trie 树节点表示的串匹配到 AC 自动机的哪个节点。可以用 bfs/dfs 遍历 Trie 树。

其他解法

本题的三种解法