字符串算法

北京理工大学 林恺

2021 年 7 月 12 日

目录

字符串 hash

快速幂

```
ll qpow(ll bsc,ll y){
    ll ret = 1;
    while(y){
        if(y&1) ret = ret*bsc%mod;
        bsc = bsc*bsc%mod;
        y >>= 1;
    }
    return ret;
}
```

费马小定理

对于任意质数 p，底数 $x(2 \leq x \leq p-1)$，有：

$$x^{p-1} \equiv 1(mod\ p)$$

因此在模 p 意义下，**除以 x 等价于乘上 x^{p-2}**

什么是 hash

hash 是一种将字符串转化为数字的对应关系，用于加快字符串比较的速度。

但这种关系并不是一一对应的，因此有 **hash 冲突**。

基本的 hash 方法

最常见的 hash 方法：

$$f(s) = \sum s_i \times base^i \mod p$$

其中：

- s_i 是字符串的第 i 位字母对应的一个**非零**数，例如 s 是一个小写字母组成的串时，可以采用 s[i]-'a'+1
- p 是一个质数，如 $998244353, 10^9 + 7$ 等

如何求一个子串的 hash 值

如何求一个子串的 hash 值?

$$\frac{pre_r - pre_{l-1}}{base^{l-1}}$$

一道简单的习题

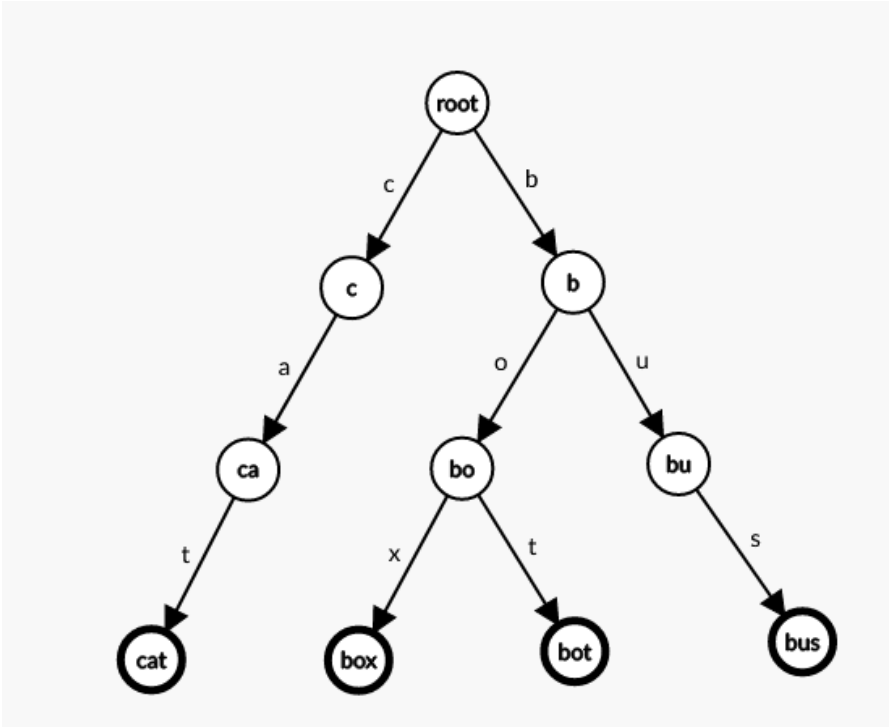
给定两个由小写字母构成的字符串 S, T，问 T 在 S 中出现了多少次?

$|S|, |T| \leq 10^5$

Trie

Trie

Trie 是一棵用来存储许多字符串的树



Trie

它很好理解

并且也很好写

```
struct node{
    int cnt;
    int ch[26];
};
```

而且还很有用

Reverse

给定 n 个数 a_i，你需要完成以下两种操作共 q 次：

- 给定 k，对于所有比 k 大的数，做“Selia 翻转”。Selia 翻转是指将 k 与被翻转的数写成二进制形式，找到他们最高的不同的位，翻转低于该位的所有二进制位（0 变成 1，1 变成 0）。
- 询问值在 $[l, r]$ 内的数有几个。

$n, q \leq 10^5$ $a_i \leq 10^9$

Reverse

对这些数建 01Trie，即把这些数写成二进制形式，再将其作为字符串插入到 Trie 中。

联系昨天所学的 lazytag ，发现第一个操作可以通过打 tag 很轻松的实现，第二个操作实质上就是一个区间求和，可以模仿线段树的方法实现。

KMP

问题的引入

再看一遍这个问题，还有别的更快的做法吗?

给定两个由小写字母构成的字符串 S, T，问 T 在 S 中出现了多少次?

$|S|, |T| \leq 10^5$

KMP

设 $S = abcxabcxabcy$, $T = abcxabcy$

注意到，无论何时，只要成功匹配到了 T 串的前 7 个字母 $(abcxabc)$，就一定能匹配到 T 串的前 3 个字母 (abc)

KMP

把上述性质一般化，设当成功匹配到 T 串的前 k 个字母时，总能匹配到 T 串的前 $next_k$ 个字母，这就是 KMP 中的 $next$ 数组。

例如，对于 $T = abcxabcy$ ， $next$ 数组为 $\{0, 0, 0, 0, 1, 2, 3, 0\}$

KMP

利用 $next$ 数组，每当匹配失败时，设 T 串上已经匹配了 k 个字母，则改为 T 串上已经匹配了 $next_k$ 个字母，再尝试进行匹配。

此时，在 $next$ 数组已经求好的情况下，对 S 串进行匹配的复杂度为 $O(n)$ 。

KMP

如何求 next 数组?

跟自己做匹配!

总复杂度 $O(n)$

最小循环节

给定字符串 S，你可以在 S 末尾加上任意字符串 T，求 $S+T$ 的最小循环节。

$|S| \leq 10^5$

最小循环节

设字符串串长为 n，则其最小循环节长度为 $n - next_n$

很神奇?

仔细想想，其实很显然。

AC 自动机

问题的引入

给定一个匹配串 S 和 n 个被匹配串 T_i，求有多少串 T_i 是 S 的子串。

$|S|, \sum_{i=1}^{n} |T_i| \leq 10^5$

AC 自动机

跑 n 次 KMP？复杂度 $O(n|S|)$，不能通过。

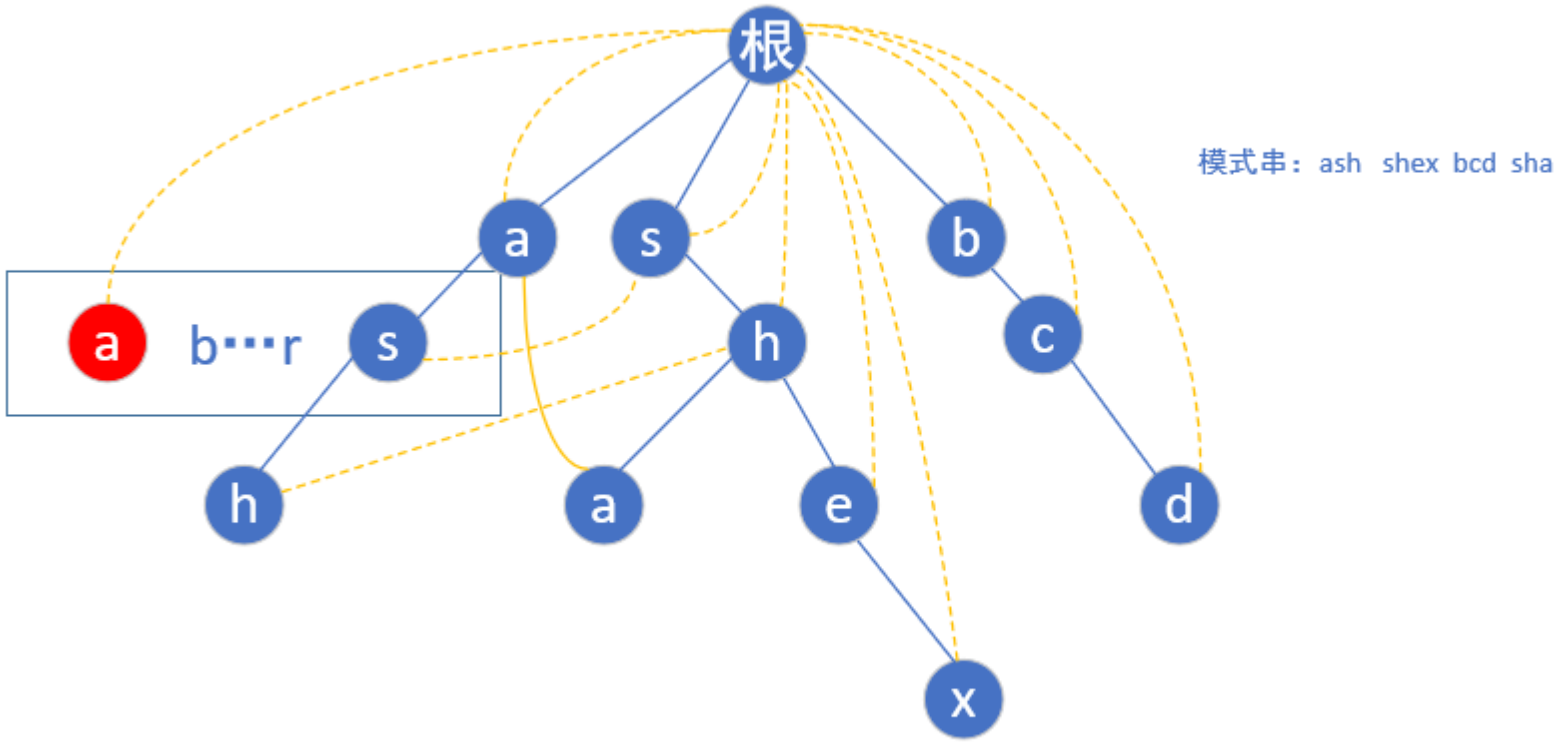
有没有什么办法一次对 n 个串同时进行 KMP？

AC 自动机

首先对这 n 个串建一个 Trie 树。

然后我们为 Trie 树上的每个节点额外加一个作用与 next 相似的 fail 指针，表示当前串匹配失败时应该从哪里开始继续匹配。

AC 自动机



AC 自动机

如何求 fail 指针?

思想与 KMP 是一致的，即通过父节点的 fail 指针得到当前节点的 fail 指针。

fail 树

观察我们建立的 fail 指针，容易发现这是一个树结构。

fail 树的性质：

- 一个节点 p 的所有子节点均包含了 p 点所代表的字符串
- 当一个字符串运行到节点 p 时，代表着点 p 到 fail 树树根上的字符串都出现了一次
- ……

剩下的部分就看你**树理**知识是否扎实了。

加一点树理知识之后……

给定一个匹配串 S 和 n 个被匹配串 T_i，每个串 T_i 在 S 中出现了多少次。

$|S|, \sum_{i=1}^{n} |T_i| \leq 10^5$

Manacher

问题的引入

给定一个字符串，请你求出其中最长的回文子串。

$|S| \leq 10^6$

Manacher

回文串问题也是竞赛中常考的一类问题

~~不过这几年好像都没怎么考过，大概是因为出不出好玩的新题来~~

所以就讲讲其中最简单的 Manacher ~~(其实是因为我不会回文树)~~

Manacher

不妨只考虑长度为奇数的回文串。(为什么可以不妨一会再说)

考虑对于每个位置，求出以该位置为中心的最长回文串，朴素的算法需要 $O(n^2)$ 的时间

通过利用回文串的对称性，可以达到加快求回文串长度的目的。

Manacher

考虑从左往右求出以每个位置为中心的最长回文串。事实上，我们求值的速度和我们访问字符串的速度是不一样的，而已经被访问过而未被求值的区域，就可以利用已经求出来的回文串来加速。

Manacher

考虑字符串 abacabac ，当我们求出以前 4 个字符为中心的最长回文串时，事实上我们已经用到了 abacaba 这 7 个字符的信息。而对于右侧的 aba ，由对称性可以知道，以第 6 个字符为中心的回文串至少长度为 3 （由第 2 个字符的信息对称过来得到)。

因此在求以第 6 个字符 b 为中心的最长回文串时，我们不必再对第 $5 \sim 7$ 个字符进行检查。同样的，求第 $5, 7$ 个字符为中心的最长回文串时也不需要用到 $5 \sim 7$ 个字符的信息。

Manacher

更一般地，只要充分利用了对称信息，即维护出利用了当前最右端信息的回文串的位置，求出以每个位置为中心的最长回文串的速度是 $O(n)$ 的。

题目

没有题目

完结撒花

祝大家下午切题愉快！

谢谢大家！