sclbgw7

Leave them underground.

AC自动机学习笔记-1（怎么造一台AC自动机?）

月更博主又来送温暖啦QwQ

今天我们学习的算法是AC自动机。AC自动机是解决字符串多模匹配问题的利器，而且代码也十分好打=w=

在这一篇博客里，我将讲解AC自动机是什么，以及怎么构建一个最朴素的AC自动机。（不知道为什么我写出来的AC自是大得要命=。=）

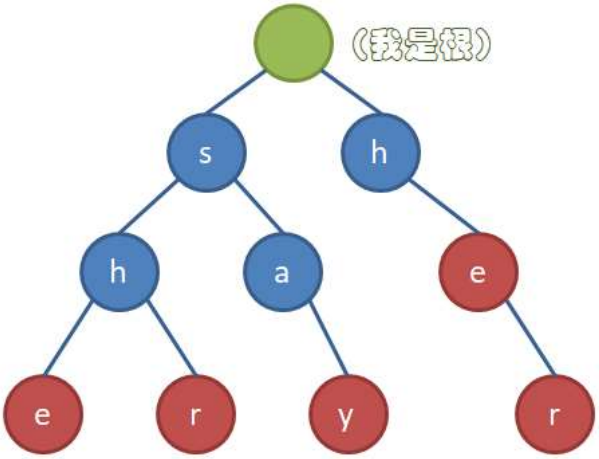
前置知识

首先你一定要对Trie树以及KMP了如指掌，尤其是要明白KMP中失配数组(next或fail数组)的本质:利用已经匹配过的部复的匹配，达到快速匹配的目的。

AC自动机是什么

大家都知道KMP可以用于在一个大字符串（文本串）中寻找另一个小的字符串（模式串），那么如果有n个模式串，要部在文本串中找出来呢？当然，我们可以做n次KMP（别小瞧30分哦），但是其效率并不能差强人意。这个时候，我们模式串做成Trie树，似乎可以提高效率。

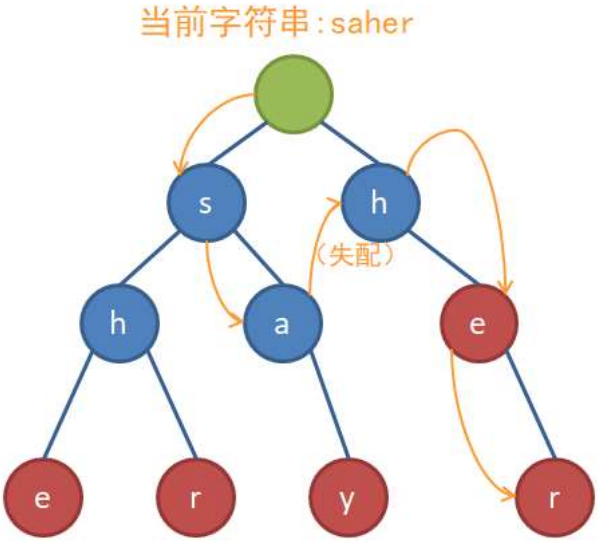
比如说，我们有5个模式串：she,shr,say,he,her,那么它们所建出来的Trie树应该是长成这样的：（红色节点表示单词



那么，怎么用它来匹配呢？如果我们把文本串的每一个点都作为起点放到Tire树上匹配，它的复杂度将会是...我要你Ti(ノ'□')ノ ┻┻

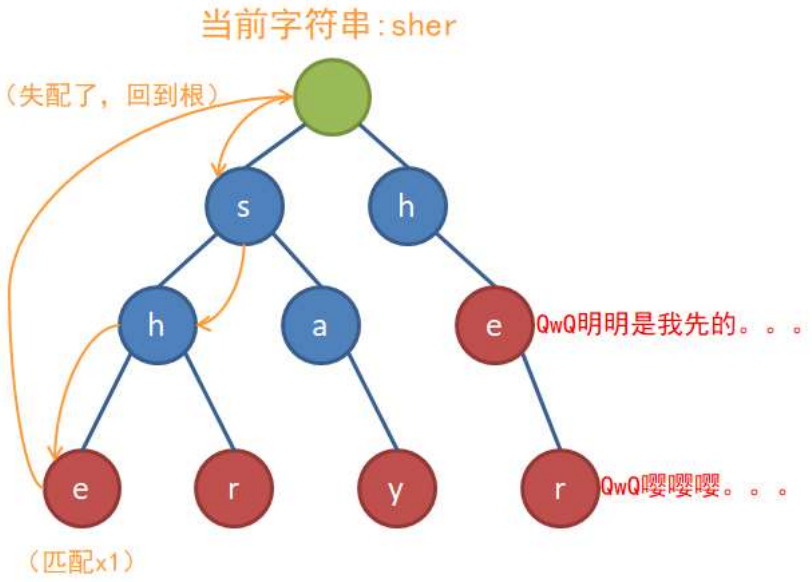
既然这样，那么如果只把文本串的第一个字符为起点，会发生什么呢？

你以为会是这样的：

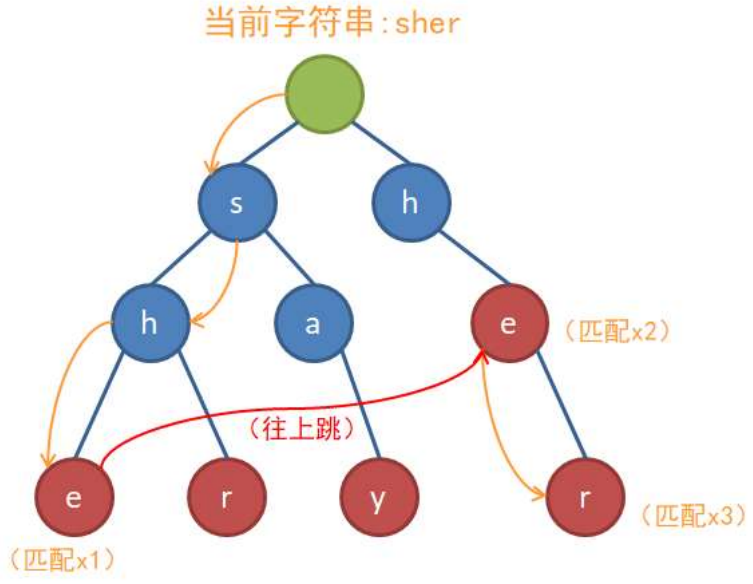


完美！

然而实际上却是这样的：



问题很明显，当我们匹配完she时，he其实也被匹配到了。所以我们希望这棵Trie树上能够加点东西，让它可以达到下

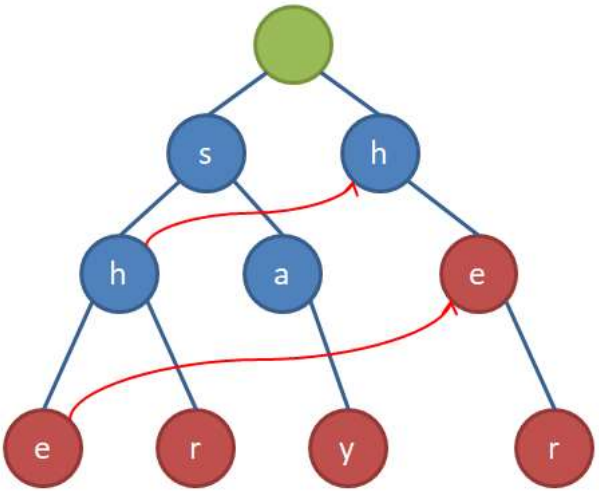


上图中，红色的箭头就是失配指针——fail指针。它表示文本串在当前节点失配后，我们应该到哪个节点去继续匹配。
每个节点，我们要找到这个节点-代表的字符串-在树上所有的节点-表示的字符串中-能找到的最长的后缀，意思就是"

了这个点，我也相当于匹配到了的节点（中的深度最大的节点）。”比如说，在我举的例子中，当我们匹配到了she时，走的路径也包含了he，he是she的一个后缀。我们在she上失配，至少说明我们已经匹配到了he，于是就可以跳到代表继续匹配。

到这里，你是不是发现fail指针和KMP中的next指针简直一毛一样？它们都被称为“失配指针”。将Trie树上的每一个点都针，它就变成了AC自动机。AC自动机其实就是Trie+KMP，它可以用来解决在文本串中寻找很多模式串，即多模匹配问

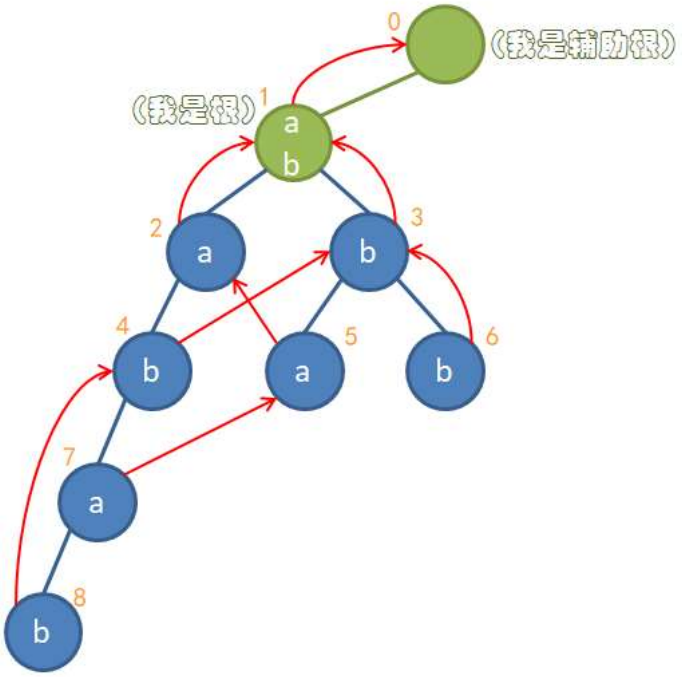
对于一开始的5个单词，它们所构建出的AC自动机就长这样（没有画出红色箭头的点，其fail指针都指向根节点）：



如何构建AC自动机

显然，我们要做的就是快速地求出所有点的fail指针。我们以bfs的顺序依次求出每个节点的fail，这样，当我们要求一个时，它的父亲的fail肯定已经求出来了。若当前节点为A，其父节点为B，B的fail为C，那么C所代表的字符串一定是B的缀。如果C有一个儿子D的字符与A的字符等同，那么显然D所代表的串（C加一个字符）就是A所代表的串（B加一个字后缀。如果C没有一个儿子，使其字符与A的字符等同呢？很简单，只需要再访问C的fail就行了。如此反复，直到A的最到，或者A的fail指向根节点为止。（A在Trie树中没有后缀，乖乖回到根重新匹配吧！）

为了解释得更清楚，我举一个例子。下面这幅图是我根据别的地方的图重新画的（n次转载？），出处我没找到_(:з」∠根据bfs序标号的。



步骤：

1. 为了少一些特判，设置一个辅助根节点0号节点，0号节点的所有儿子都指向真正的根节点1号节点，然后将1号节点的fail指向0号节点。
2. 找到2号节点的父亲节点的fail节点0号节点，看0号节点有没有为a的子节点。有，于是2号节点的fail指向1号节点。
3. 找到3号节点的父亲节点的fail节点0号节点，看0号节点有没有为b的子节点。有，于是3号节点的fail指向1号节点。
4. 找到4号节点的父亲节点的fail节点1号节点，看1号节点有没有为b的子节点。有，于是4号节点的fail指向3号节点。
5. 同上。

6. 同上。
7. 同上。
8. 找到8号节点的父亲节点的fail节点5号节点，看5号节点有没有为b的子节点。没有，于是再找到5号节点的fail节点2号节点，看2号节点有没有为b的子节点。有，于是8号节点的fail指向4号节点。

这样，一个AC自动机就做好了。

注意到由于辅助节点的存在，我们不需要做任何特判，在树上没有后缀的节点的fail指针会自动连向根节点。

构建fail指针的代码：

```
void build()
{
        for(int i=0;i<26;++i)ch[0][i]=1;
    fail[1]=0;
    queue<int>q;
    q.push(1);
    while(!q.empty())
    {
        int x=q.front();q.pop();
        for(int i=0;i<26;++i)
        {
                int c=ch[x][i];
            if(!c)continue;
            int fa=fail[x];
            while(fa&&!ch[fa][i])fa=fail[fa];
            fail[c]=ch[fa][i];
            q.push(c);
        }
    }
}
```

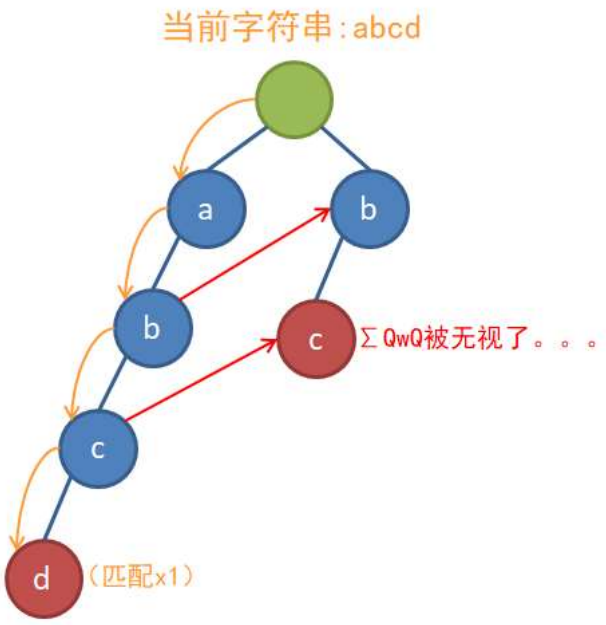
如何利用AC自动机来查找

这个问题似乎显而易见，只要根据文本串的内容沿着Trie树的边往下走就行了，一失配就沿着fail边向上跳。

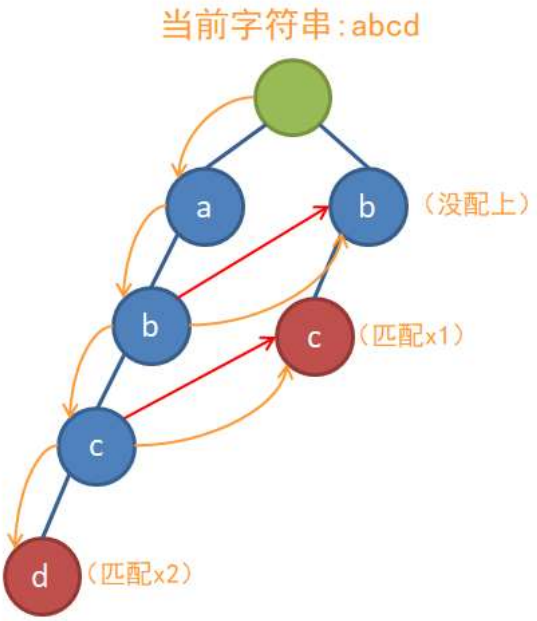
。。。

我在被大佬虐飞之前也是这么想的QwQ

fail边不只是失配指针这么简单，如果你像我刚才说的那么做的话，你就可能会面临下面这样的问题：



为了不让这种事情发生，我们每遇到一个fail指针就必须向上跳到顶，以保证不会漏过任何一个子串，就像这样：



当然，这样未免也太蠢了，于是这里又有一个小优化：如果一个节点的fail指向一个结尾节点，那么这个点也成为一个点。在匹配时，如果遇到结尾节点，就进行相应的计数处理。

进行匹配的代码：

```
void print(int x)
{
        while(x)
    {
        if(end[x])
        {
                        //计数、打印等等，视题目要求而定
        }
        x=fail[x];
    }
}

void match(char *s)
{
        int len=strlen(s),now=1;
    for(int i=0;i<len;++i)
    {
        int id=s[i]-'a';
        while(now&&!ch[now][id])now=fail[now];
        now=ch[now][id];
        if(end[now]||en[now])print(now);
        //en[now]即为伪结尾标记
    }
}

//记得在build中加上这句话
void build()
{
        ...
    if(end[fail[c]]||en[fail[c]])en[c]=1;
    ...
}
```

一个被我们忽略的问题

时间复杂度？？？

设模式串平均长度为 l ，建树复杂度为 $O(nl)$ ，构建fail指针为 $O(nl)$ ，匹配时因为每次都要跳fail边，复杂度上界i $O(ml)$ ，所以总复杂度为 $O((n+m)l)$ ！

这和暴力有什么区别(ノ°Д°)ノ ┻┻？？？

虽然说，这个上界应该是十分松的，但是我们想要的是能跑 $1e6$ 的速度！

这个时候我们就需要优化了。。。然而我已经没时间写辣QwQ!这些就留到下一篇博客吧！

谢谢你的资瓷啦QwQ!

标签： AC自动机

sclbgw7
粉丝 - 4 关注 - 1
+加关注

11

« 上一篇： Dynamic Rankings || 动态/静态区间第k小（主席树）
» 下一篇： 打FFT时中发现的卡常技巧

posted @ 2018-07-03 21:49 sclbgw7 阅读(3992) 评论(0)

关于我

名字：韦特箫
坐标HNYZ，真实全组倒数第一

友链们

OI队友
最强上仙litble
古典男神boshi&&面瘫土豪XZY（该站点已倒闭）
社会毒瘤zyf（已退役）
文学大师annoyrain
理性缜密Dimitry_I（已退役）
乖巧可爱Rayment
万堆之爷Chlience
正义帅气jiangbojun2017
外校dalao
人美歌甜Menhera
里面的花penth

昵称： sclbgw7
园龄： 5年1个月
粉丝： 4
关注： 1
+加关注

搜索

找找看
谷歌搜索

我的标签

主席树(3)
游记(2)
AC自动机(2)
状态压缩(1)
状压DP(1)
贪心(1)

树状数组(1)
FFT(1)
卡常(1)
树形dp(1)
更多

阅读排行榜

1. AC自动机学习笔记-1（怎么造一台AC自动机?）(3992)
2. AC自动机学习笔记-2（Trie图&&last优化）(2407)
3. 后缀自动机(SAM)速成手册！(1044)
4. 打FFT时中发现的卡常技巧(498)
5. 主席树初步学习笔记（可持久化数组?静态区间第k大?）(476)

评论排行榜

1. NOIP2018溅死记(3)
2. AC自动机学习笔记-2（Trie图&&last优化）(2)
3. 后缀自动机(SAM)速成手册！(1)

推荐排行榜

1. AC自动机学习笔记-1（怎么造一台AC自动机?）(11)
2. AC自动机学习笔记-2（Trie图&&last优化）(8)
3. 后缀自动机(SAM)速成手册！(1)
4. 打FFT时中发现的卡常技巧(1)

最新评论

1. Re:AC自动机学习笔记-2（Trie图&&last优化）
懂了！亏我还自己想半天找例子模拟hh
--pipipapi
2. Re:AC自动机学习笔记-2（Trie图&&last优化）
非常有用
--修落
3. Re:后缀自动机(SAM)速成手册！
论气死设计师朋友的正确姿势（逃）
--totorato
4. Re:NOIP2018溅死记
@ penth(物理)屋里系教授...
--CKnight
5. Re:NOIP2018溅死记
@ CKnight在大学养老岂不是要当教授...
--penth