

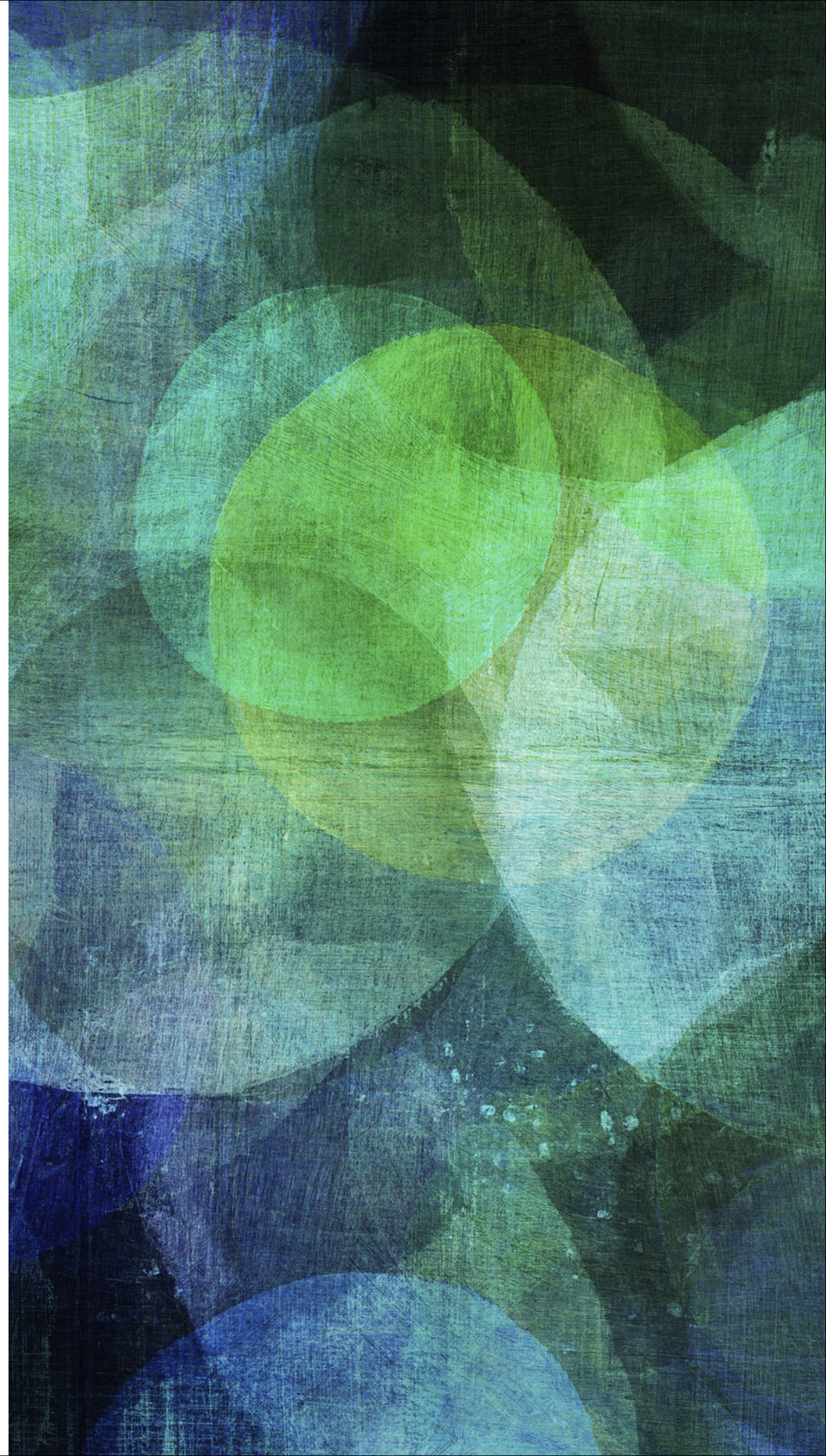
# AC自动机

---



# 字符串匹配

---

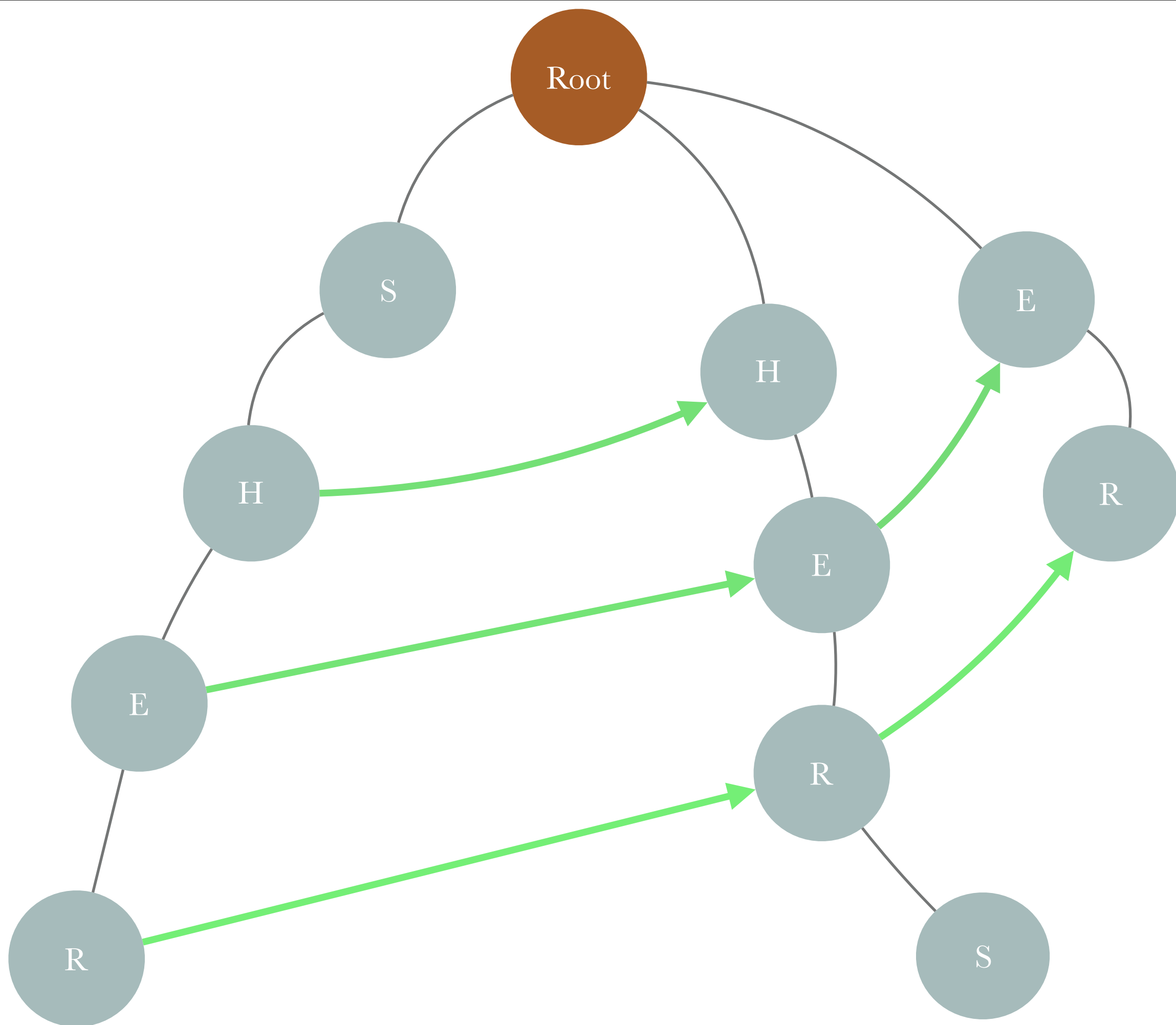


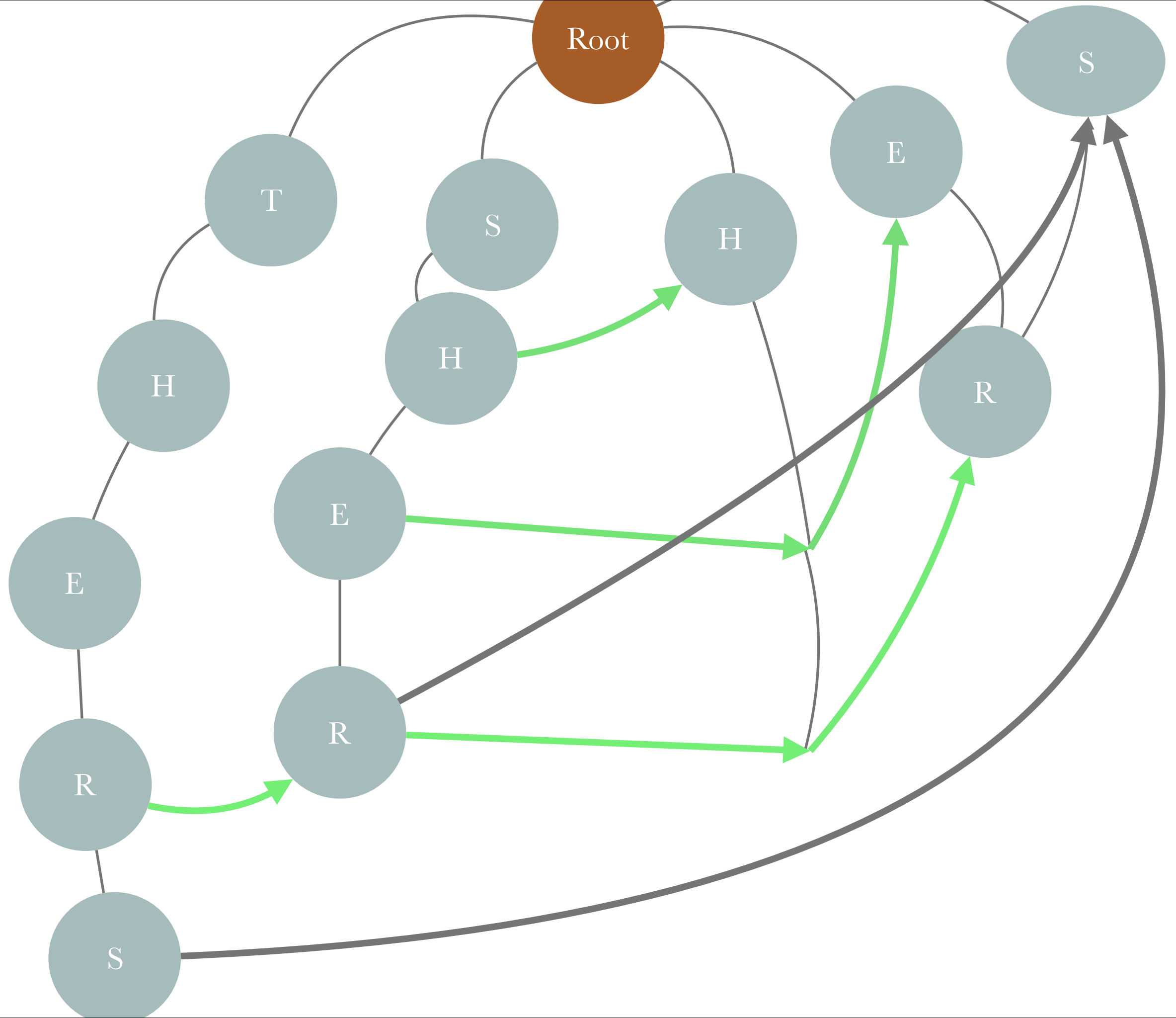


## 一个例题

---

- 你有一些模式串，有一个文本串
- 你需要在文本串中找到出现了几种模式串
- she her h
- 在sher中出现三种模式串





# 失配指针

---

- 每当发现当前节点没有要匹配的后继节点后，我们都需要找到最长的一个前缀来匹配当前的后缀去继续匹配，也就是找到从字典树的根出发的最长的一个路径满足这个路径构成的字符串是当前节点代表的前缀的一个后缀
- 对于模式串sher, hers, er
- 比如我们要查询shers这个字符串匹配了几个模式串
- 当在字典树中走完sher，我们应该找到其他字符串最长的一个前缀满足是sher的后缀，那就是hers的前缀，去继续匹配

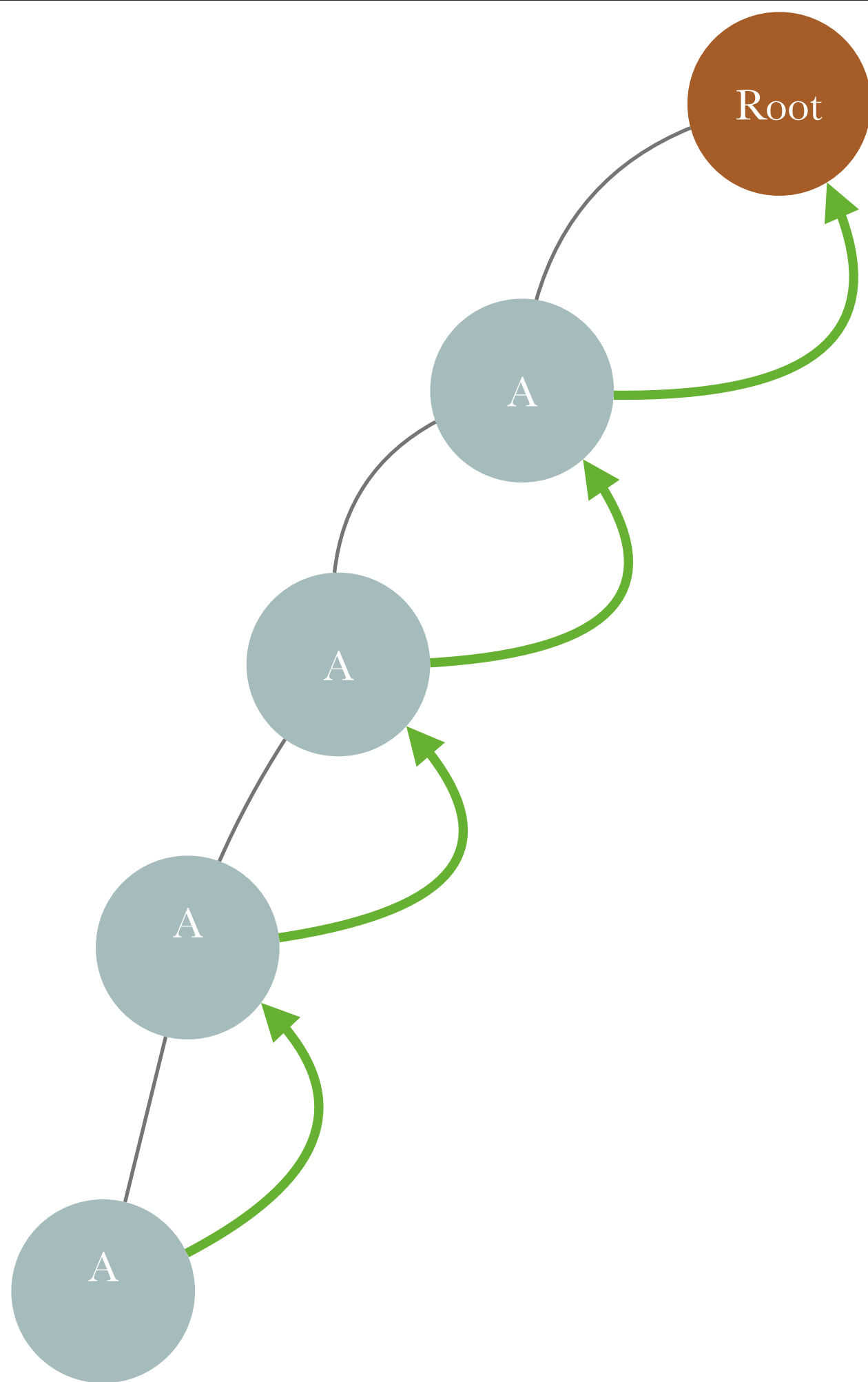
# 如何找失配指针

## 如何找失配指针

---

- $\text{fail}[0] = 0;$
- 假设当前节点 $u$ 不存在 $c$ 儿子了
- 假设父亲的 $\text{fail}$ 指针为 $\text{fail}[\text{fa}]$
- 那么当 $\text{fail}[\text{fa}]$ 有 $c$ 儿子的时候，那么 $\text{fail}[u] = \text{ch}[\text{fail}[\text{fa}]][\text{c}]$
- 否则就去看 $\text{fail}[\text{fail}[\text{fa}]}$ 有没有 $c$ 儿子
- 如果一直没有， $\text{fail}[u] = 0$ (根)
- 因此我们利用一个bfs就可以处理出所有点的 $\text{fail}$ 指针了



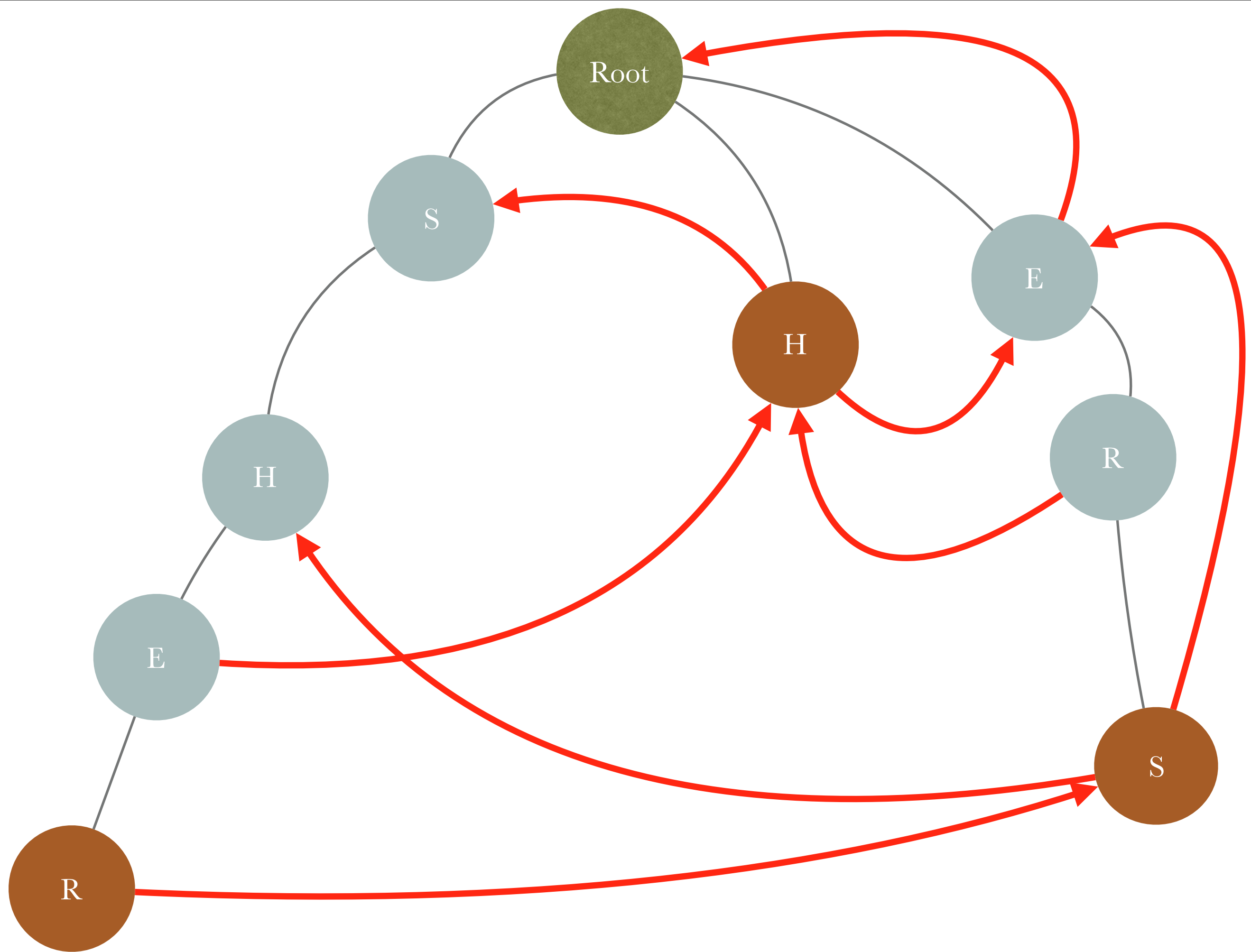


# TRIE图

# TRIE图

---

- ▶ 当一个节点 $u$ 没有某个 $c$ 儿子的时候，我们需要让 $u$ 的 $c$ 指向直接连接到 $u$ 的 $fail$ 指向的 $c$ 儿子节点上，将所有的空儿子都连上后继节点，这便是trie图
- ▶ 有了这个图，我们就可以 $O(1)$ 直接走到下一个该走到的点，不必每次都一直 $fail$ 去找
- ▶ 图论算法与字符串匹配结合可以延伸很多题目
- ▶ 比如，求不包含模式串的长度为 $len$ 的字符串的数量
- ▶  $dp[n][node]$ 表示长度为 $n$ 走到了 $node$ 节点的方案数，只要不走危险节点（模式串的尾节点）即可，枚举所有的后继节点去转移到 $dp[n+1][node\_next]$ ，问题其实完全变成了图论， $dp$ 问题



```

void Insert(char *a) {
    int p = 0;
    for(; *a; a++) {
        int c = ID[*a];
        if (!ch[p][c]) {
            memset (ch[sz], 0, sizeof(ch[sz]));
            val[sz] = 0;
            ch[p][c] = sz++;
        }
        p = ch[p][c];
    }
    val[p] = 1;
}

```

```

void Construct() {
    int *s = Q, *e = Q;
    for (int i = 0; i < CD; i++) {
        if (ch[0][i]) {
            fail[ ch[0][i] ] = 0;
            *e ++ = ch[0][i];
        }
    }
    while (s != e) {
        int u = *s++;
        for (int i = 0; i < CD ;i++){
            int &v = ch[u][i];
            if(v) {
                *e++ = v;
                fail[v] = ch[ fail[u] ][i];
            } else {
                v = ch[ fail[u] ][i];
            }
        }
    }
}

```



FAIL树

# FAIL树

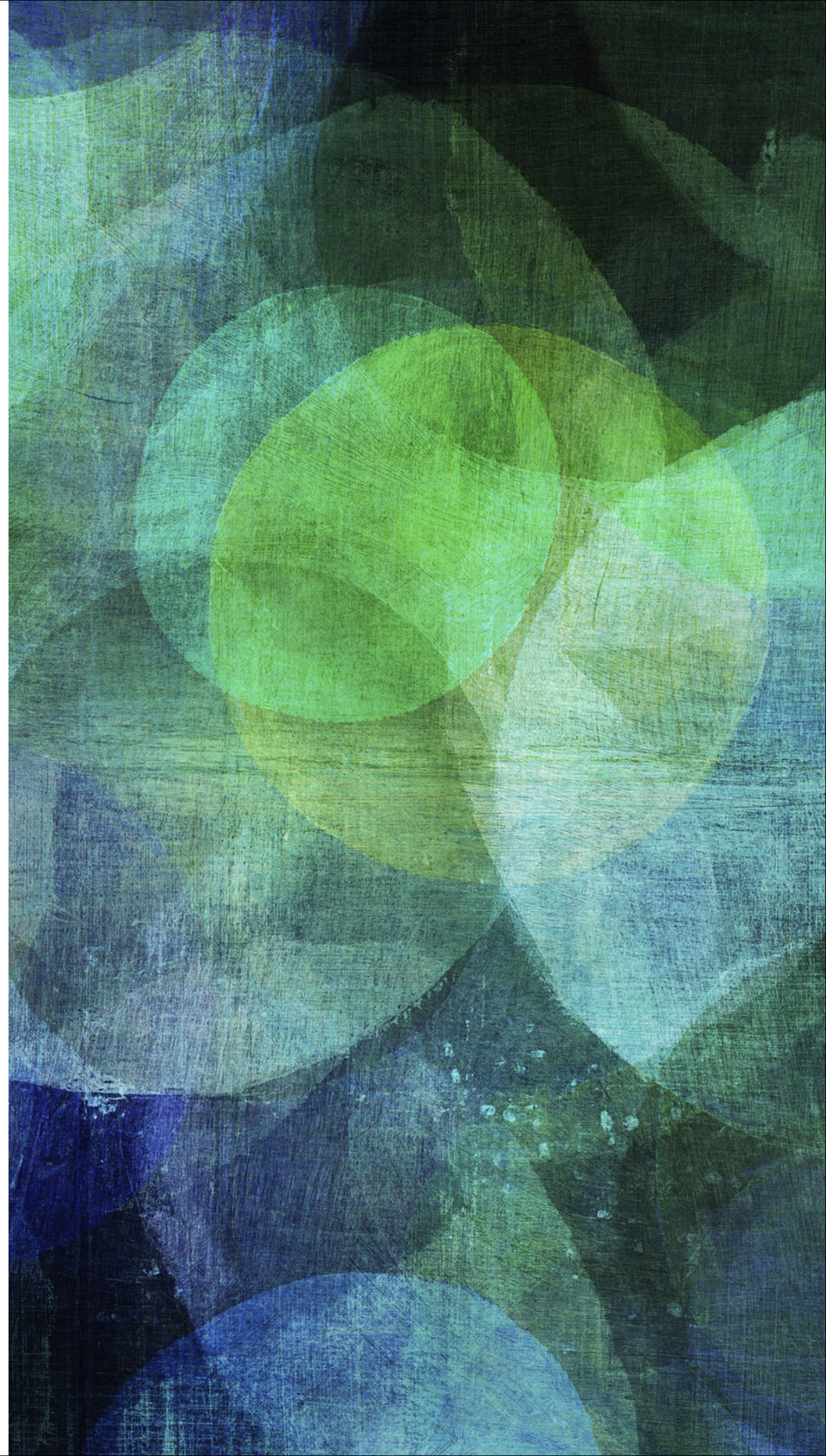
---

- 将每个点的fail与自己建立一条有向边，我们便得到了一颗fail树
- fail树是干啥的呢
- fail树上一个点到根的路径就相当于我们原先一路fail到根
- 两个节点的LCA就相当于匹配两个前缀的后缀的最长公共前缀
- 那么我们便可以支持动态插入删除某个字符串，然后查询一个字符串匹配了多少个模式串
- 查询的时候将文本串在自动机上边走边快速算出一个点fail到根的路径上的权值之和
- 查询一个点到根的权值之和可以转换成dfs序问题来做



# 后记

---





真正的AC自动机

什么是思维



你的答案？

# 我的体会

# 思维的本质

---

- 第一是与事物产生连接的能力，或者叫做洞察力
- 第二是抽象的能力
- 从图论的角度讲就是建图与缩点的能力
- 你将建好的图，不断的缩点，合并，以至于起点到终点的最短路不断的缩短
- 当你学会了一个又一个点的时候，你需要去连接这些点，没有连接与组合，我们只是一个灵活一点的移动硬盘
- 但这一切的连接与合并全是在内在完成的，外显出来的只是完成了一件事情，解决了一个问题而已
- 因此我们常常忽略了占据绝大多数比重的连接能力
- 而把注意力错放到了具体的解决方案上

- 每当我们面对一个自己不懂的问题，面临一种困境的时候，恰恰是连接能力开始生根发芽的时候，我们会穷尽我们所有的一切去分析问题，解决问题
- 在各种尝试都失败的时候，你就明白哪些连接是行不通的，即哪些边是多余的，不用建的
- 这样我们就可以在一次又一次的思维磨合中最小化我们的思维有限状态自动机
- 但这个时候如果随意的一HE解千愁，其实浪费了很好的一次机会
- 因此，每碰到一个自己不会的题，都可以用来检测自己哪一个方面的连接能力弱了，使劲反思，一路dfs到底，把没连接的部分给接上，使自己的思维体系渐渐形成一个系统

“

破山中贼易，破心中贼难

-王阳明



还有

一定要敢想

想完后加踏实地  
去干

- 
- 上海交通大学林晨曦，名言：我们得不到冠军，不是我们做不到，而是不敢想。主动留级备战全球总决赛，随后交大拿到了中国高校历史上第一个world final 冠军，也是亚洲第一个总冠军，迄今为止大陆的三个总冠军都是交大夺得
  - [http://science.china.com.cn/2016-11/25/content\\_9182185.htm](http://science.china.com.cn/2016-11/25/content_9182185.htm)