

# SAM

```

struct nodes {int len, fa, ch[26]};
//当前节点最长 当前节点父节点（最短子串没有第一位）树 当前节点子节点（最长子串后面加字母）DAG
vector<nodes> node(n * 2 + 10);
int last = 1, tot = 1;
//最后一个节点 一共有多少个节点
auto extend = [&](int c) -> void {
    int p = last, np = last = ++ tot;
    node[np].len = node[p].len + 1;
    for (; p && !node[p].ch[c]; p = node[p].fa) node[p].ch[c] = np;
    if (!p) node[np].fa = 1;
    else
    {
        int q = node[p].ch[c];
        if (node[q].len == node[p].len + 1) node[np].fa = q;
        else
        {
            int nq = ++ tot;
            node[nq] = node[q], node[nq].len = node[p].len + 1;
            node[q].fa = node[np].fa = nq;
            for (; p && node[p].ch[c] == q; p = node[p].fa) node[p].ch[c] = nq;
        }
    }
    return ;
};
for(auto v : s) extend(v - 'a');

```

后缀自动机，从起点出发可以遍历通过 $ch$ 可以遍历字符串中所有的子串，形成 $DAG$ 。

某个节点 $fa$ 代表着这个节点的最短的串去掉首位之后所在的节点，形成树形结构。

$len$ 代表当前节点最长串。

每个路径对应一条子串。