Aho-Corasick – The Complete Algorithm

Step 1. Build a trie from pattern strings

Step 2. Add suffix links to the trie

Step 3. Add output links to the trie

Step 4. Execute find() method with an input text

Step 2 – Add Suffix Links to the Trie

```
1  suffix link of the root is nullptr.
2
3  For each child i of the root
4      suffix link of root->child[i] is the root.
5
6  Visit each node w of the trie in "level-order" (except the root)
7
8      For each non-null child a of node w:
9
10         /*** Create suffix link for node wa ***/
11
12         Let node x be the suffix link of w
13
14         While (node x is not nullptr) and (x has not child a):
15             x = suffix link of x;
16
17         If x is nullptr:
18             suffix link of node wa = root of the trie
19         Else:
20             suffix link of node wa = child a of x
21
22         buildOutputLink(node wa);
23
```

After adding a suffix link, lets' also add its output link!

Step 3 – Add Output Links to the Trie

Adding an output link for a node *u*

```
1  Let v be the node pointed by suffixLink of node u
2
3  If v is not null and v corresponds to a pattern:
4      outputLink of node u will point to v.
5
6  Otherwise, If v has an output link:
7      outputLink of node u will point to the outputLink of v.
8
```

Check the endOfWord flag of *v*.

Notice that this pseudo-code implies that some Nodes may not have an output link. This is ok.

Step 4 – Execute the find() method with an input text

Find occurrences inside the text,
of the patterns stored in the trie

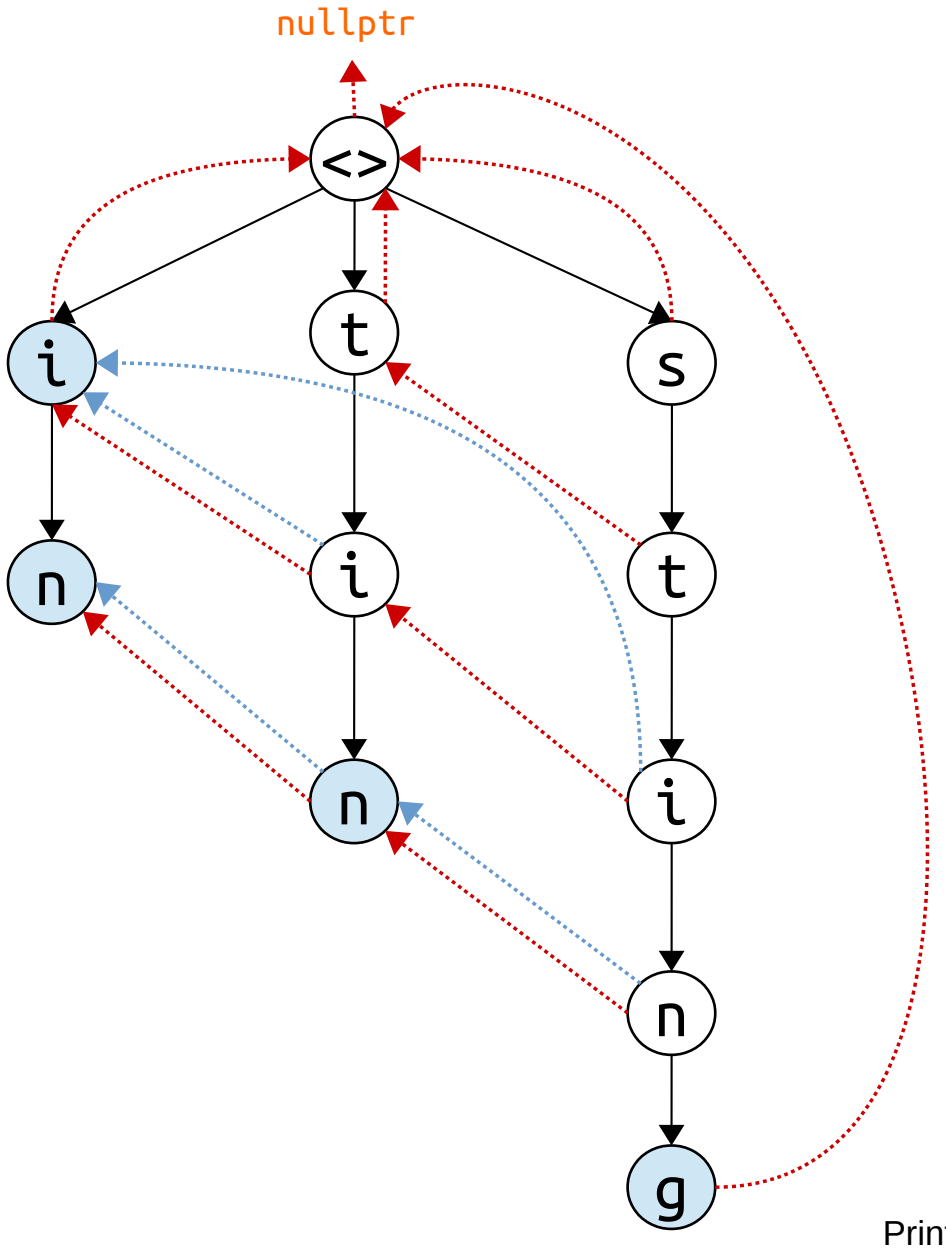
```
1   void find(std::string text)
2   {
3        Let currentNode be initially the _root;
4
5        For each character a of the text:
6
7             While (the currentNode hasn't child with letter a) And (the currentNode IS NOT the root):
8                  The currentNode will be now to the node pointed by his suffixLink
9
10            If (the current node has not child a And the currentNode IS the root):
11                 Move to the next iteration of the For loop
12
13            The currentNode will be now the child with letter a of the currentNode
14
15            If the currentNode has the flag endOfWord equal to true:
16                 // We have found a pattern in position ??? of the text! Print position and pattern
17
18            Let v be the node pointed by the outputLink of the currentNode
19                 (or nullptr if the currentNode hasn't an outputLink)
20
21            While v is not nullptr:
22                 // We have a found a pattern in position ??? of the text! Print position and pattern
23                 v will be now the outputLink of v (or nullptr if v does not have an outputLink)
24  }
```

???:How we could get the position where the
text occurred in the text?

Example

patterns:

i				
i	n			
t	i	n		
s	t	i	n	g



Print the position where every pattern occurs in the text.

```
std::vector<std::string> patterns = { "i", "in", "tin", "sting" };

AhoCorasickTrie myTrie(patterns);

myTrie.find("sting");
```

```
2 i
1 tin
2 in
0 sting
```