6.4.2 Applications of Suffix Tree

Assuming that a Suffix Tree[4] for a string `S` is already built, we can use it for these applications:

Exact String Matching in $O(|Q| + occ)$

With Suffix Tree, we can find all (exact) occurrences of a query string Q in S in $O(|Q|+occ)$ where $|Q|$ is the length of the query string Q itself and occ is the total number of occurrences of Q in S – no matter how long the string S is. When the Suffix Tree is already built, this approach is *faster* than many exact string matching algorithms (e.g. KMP).

With Suffix Tree, our task is to search for the vertex x in the Suffix Tree which represents the query string Q. This can be done by just one root to leaf traversal that follows the edge labels. Vertex with path-label $= Q$ is the desired vertex x. Then, leaves in the subtree rooted at x are the occurrences of Q in S. We can then read the starting indices of such substrings that are stored in the leaves of the sub tree.

For example, in the Suffix Tree of `S = 'acacag$'` shown in Figure 6.2, right and `Q = 'aca'`, we can simply traverse from root, go along the edge label 'a', then the edge label 'ca' to find vertex x with the path-label 'aca' (follow the dashed red arrow in Figure 6.2, right). The leaves of this vertex x point to index 1 (substring: `'acacag$'`) and index 3 (substring: `'acag$'`).

Exercise: Now try to find a query string `Q = 'ca'` and `Q = 'cat'`!

Finding Longest Repeated Substring in $O(n)$

With Suffix Tree, we can also find the longest repeated substring in S easily. The deepest internal vertex X in the Suffix Tree of S is the answer. Vertex X can be found with an $O(n)$ tree traversal. The fact that X is an internal vertex implies that it represent more than one suffixes (leaves) of string S and these suffixes shared a common prefix (repeated substring). The fact that X is the deepest internal vertex (from root) implies that its path-label is the longest repeated substring.

For example, in the Suffix Tree of `S = 'acacag$'` shown in Figure 6.2, right, the longest repeated substring is 'aca' as it is the path-label of the deepest internal vertex.

Exercise: Find the longest repeated substring in `S = 'cgacattacatta$'`!

Finding Longest Common Substring in $O(n)$

The problem of finding the Longest Common **Substring** (not Subsequence)[5] of two **or more** strings can be solved in linear time with Suffix Tree. Consider two strings $S1$ and $S2$, we can build a **generalized Suffix Tree** for $S1$ and $S2$ with two different ending markers, e.g. $S1$ with character '#' and $S2$ with character '$'. Then, we mark each internal vertices with have leaves that represent suffixes of *both* $S1$ and $S2$ – this means the suffixes share a common prefix. We then report the deepest marked vertex as the answer.

For example, with `S1 = 'acgat#'` and `S2 = 'cgt$'`, The Longest Common Substring is `'cg'` of length 2. In Figure 6.3, we see the root and vertices with path-labels 'cg', 'g', and 't' all have two different leaf markers. The deepest marked vertex is 'cg'. The two suffixes `cgat#` and `cgt$` share a common prefix 'cg'.

[4] As Suffix **Tree** is more compact than Suffix **Trie**, we will concentrate on Suffix **Tree**.

[5] In 'abcdef', 'bce' (skip character 'd') is subsequence and 'bcd' (contiguous) is substring and also subsequence.