

### Algorithm for Circular DNA alignment:

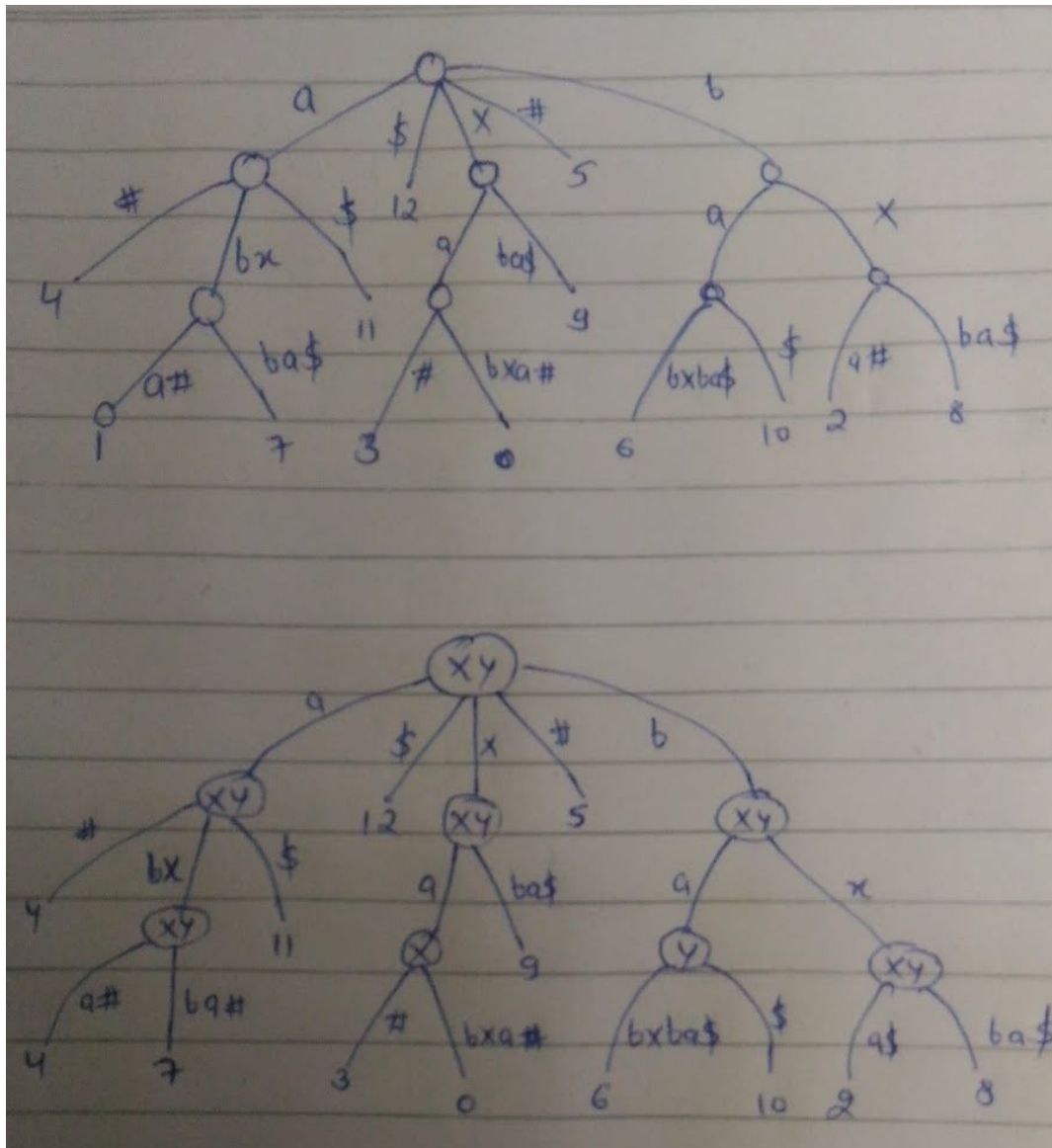
Suppose we have two circular strings to be matched of length  $m$  and  $n$ . Now we use the suffix tree for finding the rotation of each.

1>Let's say string 1 is:  $xabxa$  and string 2 as:  $babxba$ . We make a single string out of the two by separating them via  $\#$  character such that the final string to be put in suffix tree is  $xabxa\#babxba$ . We then obtain the following suffix tree for both the strings to be compared.

2>Later we modify our substring with label 'XY', 'X', 'Y' representing that the string belong to X and Y both or just Y or just X.

3>Considering this we traverse the entire suffix tree and find which all subsequences belong to both the strings.

4>Now we can generate the alignment by adding gap for the unmatched and get the final matching result of the alignment of all the possible rotations of circular strings using a single suffix tree.



**Complexity:**

For generating the suffix tree:  $O(m+n)$  //for all the rotations of each

For finding the subsequences:  $O(\max(m,n))(m+n)$  which is either  $O(n^2)$  or  $m^2$  or  $m*n$