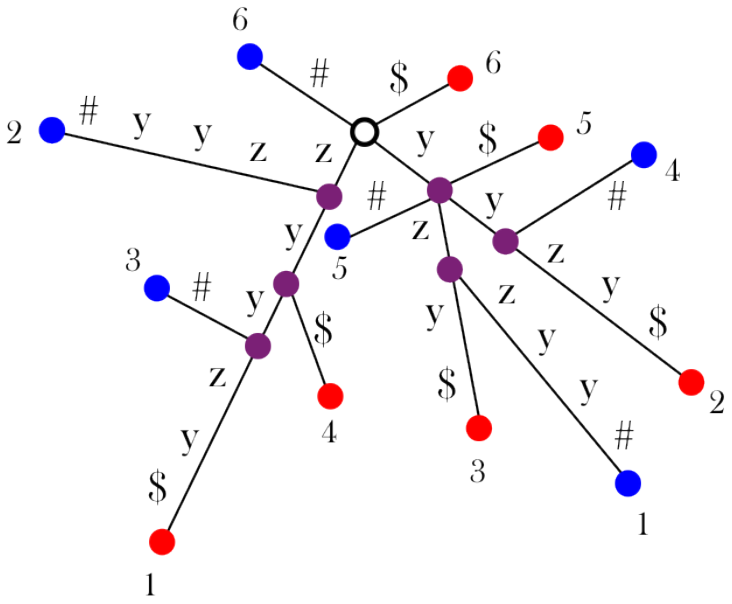
1. Build a suffix array for the string $S = zyyxzy$.

   Solution:

suffix	lex rank	i	$A_s[i]$
zyyxzy$	7	1	7
yyxzy$	5	2	4
yxzy$	4	3	6
xzy$	2	4	3
zy$	6	5	2
y$	3	6	5
$	1	7	1

2. Find the longest common substring of $zyyzy$ and $yzzyy$ using suffix trees.

   Solution: Build a generalized suffix tree for the strings $zyyzy\$$ and $yzzyy\#$. The leaves whose suffix started in the first string are colored red, and the leaves whose suffix started in the second string are colored blue. Then for each internal vertex, if it has descendants of both colors, it is colored purple. The path from the root (white vertex) to the deepest purple vertex is the longest common substring. In this case it is zyy.



3. [Gusfield, Section 1.6, Problem 3] Let α and β be strings with $|\alpha| = m$ and $|\beta| = n$. Describe an $O(m+n)$ algorithm that finds the longest suffix of α that exactly matches a prefix of β. For example, given $\alpha = ACCGTG$ and $\beta = GTGACCAGAT$, the solution is GTG.

   Solution: (sketch) One solution would be to build a suffix tree for α [runtime $O(m)$], then beginning at the root, start searching for β. Whenever we encounter a \$ during the search (meaning we've reached the end of α, although the \$ will not match the current character in β), we've found a matching prefix of β. The final such \$ we find marks the longest suffix of α that is a prefix of β.

   Searching for a prefix of β takes $O(n)$ time. If at any point we can no longer match a prefix of β, but haven't reached a \$, then that means there is a prefix of β that is a *substring* of α, but not a *suffix*, so the only string that satisfies our criteria is the empty string.

4. Let α and β be strings with $|\alpha| = |\beta| = n$. We say that α is a cyclic rotation of β if a cyclic permutation of the letters of α match β. For example, $\alpha = ACCGTG$ is a cyclic rotation of $\beta = GTGACC$.

   (a) [Gusfield, Section 1.6, Problem 1] Use *linear-time exact matching* to describe an $O(n)$ algorithm for determining whether α is a cyclic rotation of β.

   (b) [Gusfield, Section 1.6, Problem 2] Now let $|\alpha| = m$ and β be a *circular* string with $|\beta| = n > m$. A circular string of length n is a string in which character n is considered to precede character 1. Use *linear-time exact matching* to describe an $O(n)$ algorithm to find all occurrences of α in the circular string β.

   Solution:

   (a) (sketch) One solution is to build a suffix tree for β followed by β again, except without a \$ separating them. Then we search for α within this suffix tree, since each possible cyclic rotation of β is now a suffix of this augmented string. If there is a path through this suffix tree that spells out α, then α is indeed a cyclic permutation of β (since we know $|\alpha| = |\beta|$).

   (b) (sketch) In a similar fashion, you can build a suffix tree for β plus the first $|\alpha| - 1$ characters of β repeated. This ensures that if α spans part of the end and beginning of β, it will be found, but we will not return duplicate matches. We can then find all instances of α in the suffix tree just like with any pattern.