Example 2

- String1: ABRGQSE
- String2: YBRGEQUS

- Find the distance.

Tree Edit Distance

- Tree edit distance between two trees *A* and *B* (*labeled ordered rooted trees*) is the cost associated with the minimum set of operations needed to transform *A* into *B*.
- The set of operations used to define tree edit distance includes three operations:
  - node removal,
  - node insertion, and
  - node replacement.

  A cost is assigned to each of the operations.

Simple Tree Matching (STM)

- Let A and B be two trees, and i ∈ A and j ∈ B be two nodes in A and B respectively.
- A matching between two trees is defined to be a mapping M such that, for every pair (i, j) ∈ M where i and j are non-root nodes, (parent(i), parent(j)) ∈ M.
- A maximum matching is a matching with the maximum number of pairs.

Let $A = R_A{:}\langle A_1, \ldots, A_k\rangle$ and $B = R_B{:}\langle B_1, \ldots, B_n\rangle$ be two trees, where R_A and R_B are the roots of A and B, and A_i and B_j are the ith and jth first-level sub-trees of A and B respectively. Let $W(A, B)$ be the number of pairs in the maximum matching of trees A and B. If R_A and R_B contain identical symbols, the maximum matching between A and B (i.e., $W(A, B)$) is $m(\langle A_1, \ldots, A_k\rangle, \langle B_1, \ldots, B_n\rangle) + 1$, where $m(\langle A_1, \ldots, A_k\rangle, \langle B_1, \ldots, B_n\rangle)$ is the number of pairs in the maximum matching of $\langle A_1, \ldots, A_k\rangle$ and $\langle B_1, \ldots, B_n\rangle$. If $R_A \neq R_B$, $W(A, B)) = 0$. Formally, $W(A, B)$ is defined as follows:

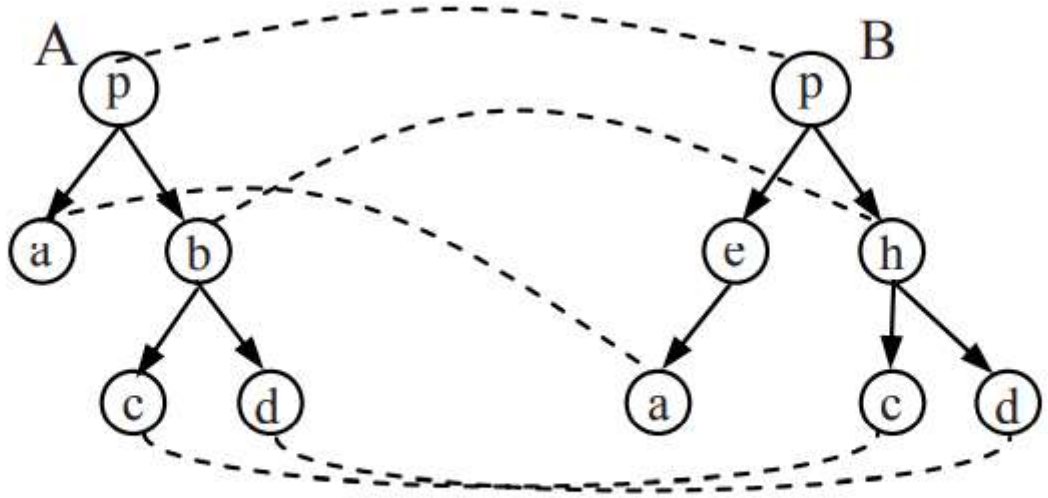
$$W(A,B) = \begin{cases} 0 & \text{if } R_A \neq R_B \\ m(\langle A_1,\ldots,A_k\rangle,\langle B_1,\ldots,B_n\rangle)+1 & \text{otherwise} \end{cases};$$

$m(\langle\rangle, \langle\rangle) = 0$ // $\langle\rangle$ represents an empty sub-tree list.

$m(s, \langle\rangle) = m(\langle\rangle, s) = 0$ // s matches any non-empty sub-tree list

$$m(\langle A_1, \ldots, A_k\rangle, \langle B_1, \ldots, B_n\rangle) = \max(m(\langle A_1, \ldots, A_{k-1}\rangle, \langle B_1, \ldots, B_{n-1}\rangle) + W(A_k, B_n),$$
$$m(\langle A_1, \ldots, A_k\rangle, \langle B_1, \ldots, B_{n-1}\rangle),$$
$$m(\langle A_1, \ldots, A_{k-1}\rangle, \langle B_1, \ldots, B_n\rangle)).$$

- Let X be a tree and let X[i] be the i^{th} node of tree X in a preorder walk of the tree.
- A mapping M between a tree A of size n_1 and a tree B of size n_2 is a set of ordered pairs (i, j), one from each matched node, satisfying the following conditions for all (i_1, j_1), (i_2, j_2),… $\in$ M:
  1. $i_1 = i_2$ iff $j_1 = j_2$;
  2. $A[i_1]$ is on the left of $A[i_2]$ iff $B[j_1]$ is on the left $B[j_2]$;
  3. $A[i_1]$ is an ancestor of $A[i_2]$ iff $B[j_1]$ is an ancestor of $B[j_2]$.



A general tree mapping example

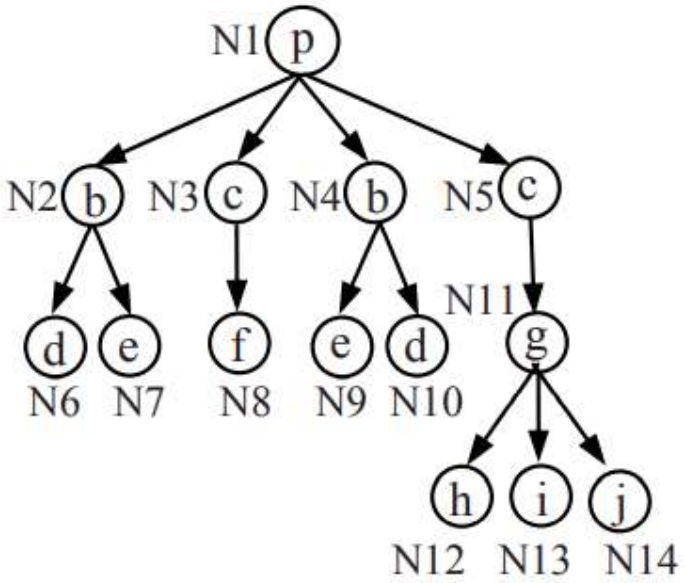
Simple tree matching

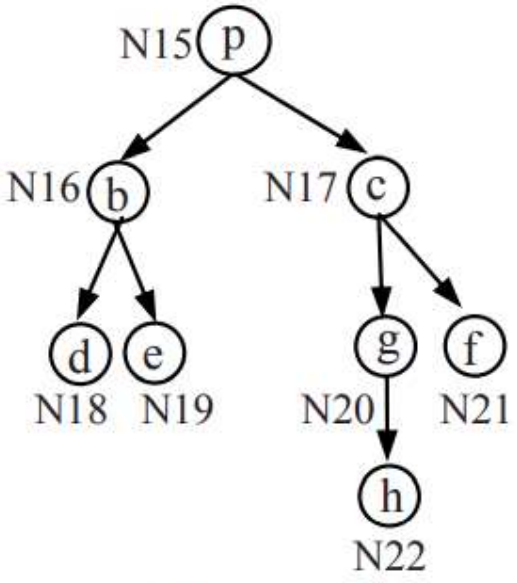
- In the general setting,
  - mapping can cross levels, e.g., node *a* in tree *A* and node *a* in tree *B*.
  - Replacements are also allowed, e.g., node *b* in *A* and node *h* in *B*.
- We describe a restricted matching algorithm, called **simple tree matching** (STM), which has been shown quite effective for Web data extraction.
  - STM is a top-down algorithm.
  - Instead of computing the edit distance of two trees, it evaluates their similarity by producing the maximum matching through dynamic programming.

Simple Tree Matching algo

Algorithm: STM(A, B)

1. **if** the roots of the two trees A and B contain distinct symbols **then**
2. **return** (0)
3. **else** $k \leftarrow$ the number of first-level sub-trees of A;
4. $n \leftarrow$ the number of first-level sub-trees of B;
5. Initialization: $m[i, 0] \leftarrow 0$ for $i = 0, \ldots, k$;
   $m[0, j] \leftarrow 0$ for $j = 0, \ldots, n$;
6. **for** $i = 1$ to k **do**
7. **for** $j = 1$ to n **do**
8. $m[i, j] \leftarrow \max(m[i, j-1], m[i-1, j], m[i-1, j-1]+W[i, j])$,
   where $W[i, j] \leftarrow \text{STM}(A_i, B_j)$
9. **end-for**
10. **end-for**
11. **return** $(m[k, n]+1)$
12. **end-if**



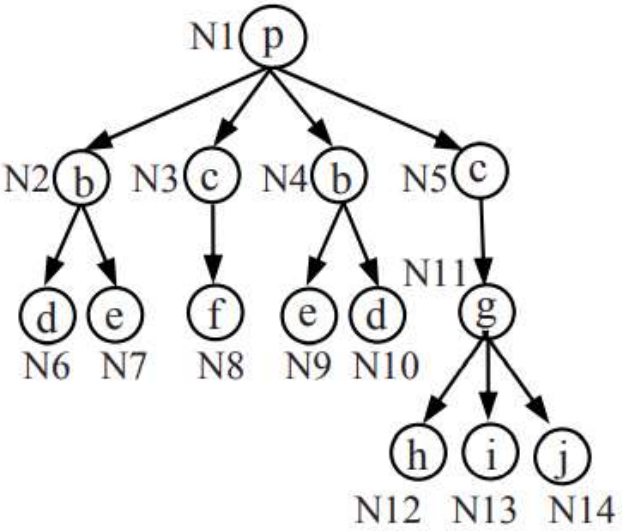


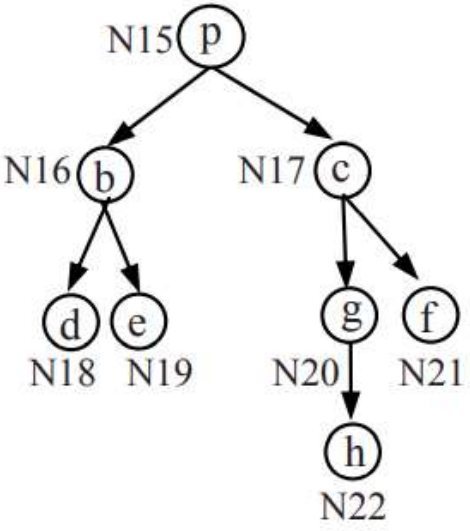
$m_{1,15}$

	0	1 (N16)	2 (N16-N17)
0	0	0	0
1 (N2)	0	3	3
2 (N2-N3)	0	3	5
3 (N2-N4)	0	3	5
4 (N2-N5)	0	3	6

$W_{1,15}$

	1 (N16)	2 (N17)
1 (N2)	3	0
2 (N3)	0	2
3 (N4)	2	0
4 (N5)	0	3





$m_{5,17}$

	0	1 (N20)	2 (N20-N21)
0	0	0	0
1 (N11)	0	2	2

$W_{5,17}$

	1(N20)	2(N21)
1 (N11)	2	0

$m_{11,20}$

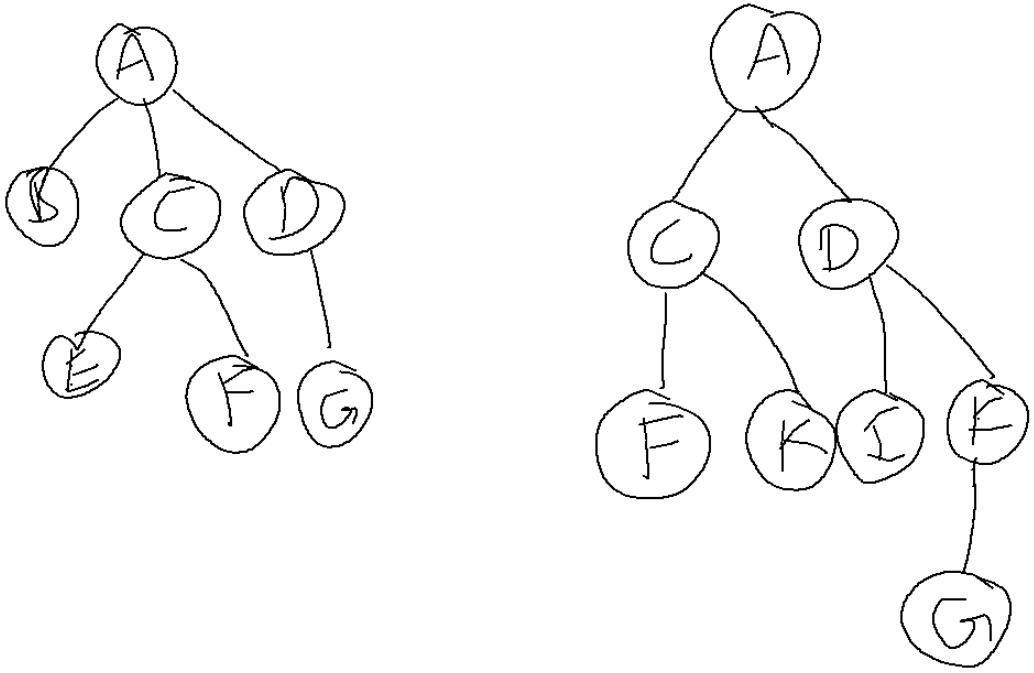
	0	1 (N22)
0	0	0
1 (N12)	0	1
2 (N12-N13)	0	1
3 (N12-N14)	0	1

$W_{11,20}$

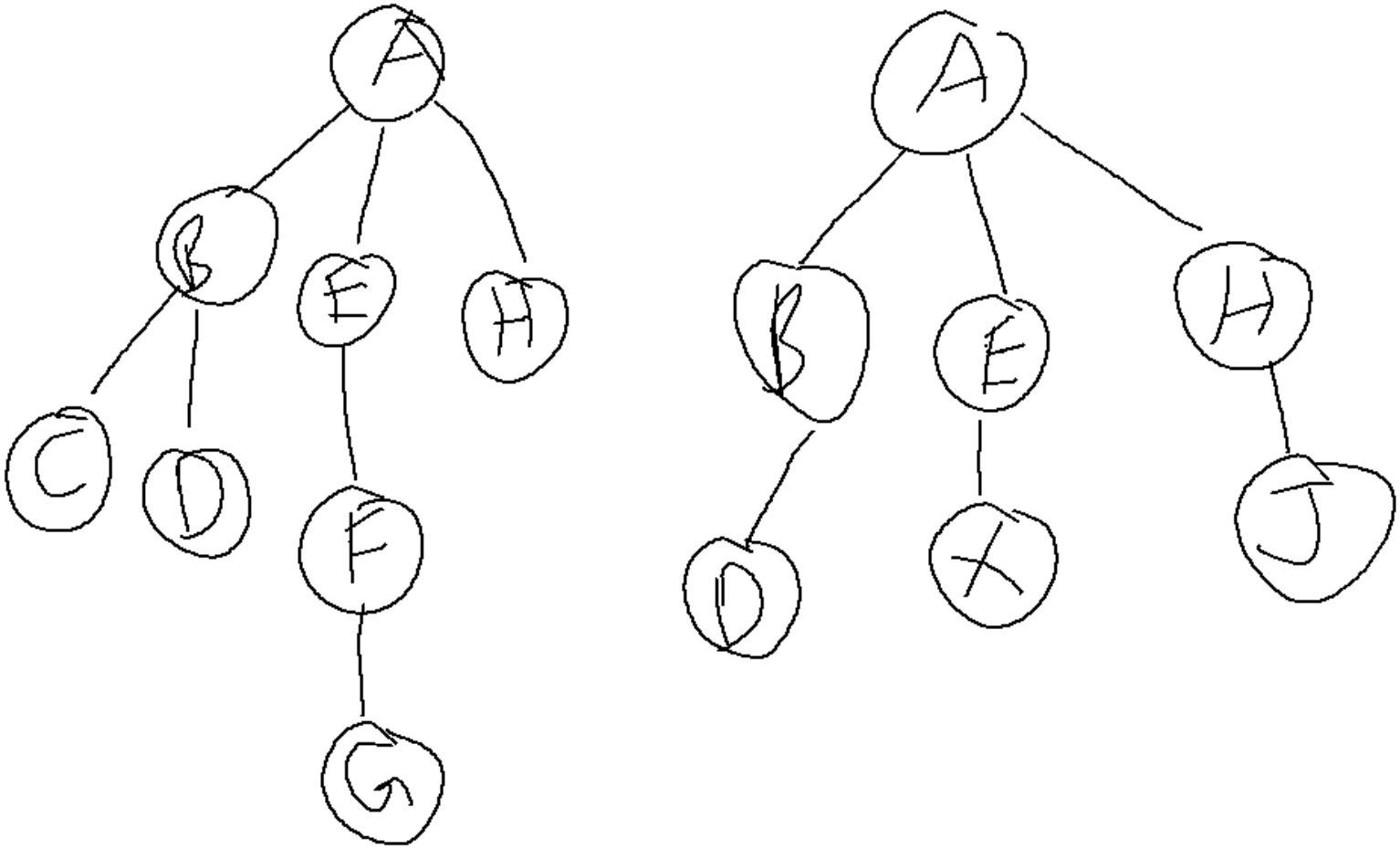
	1 (N22)
1 (N12)	1
2 (N13)	0
3 (N14)	0

Example

- Find similarity between the following trees



Find similarities between the following trees



Time and Space Complexities

- Comparison of two strings
  - Time complexity: $O(|s_1||s_2|)$
  - Space complexity: $O(|s_1||s_2|)$
- Comparison of two trees
  - Time complexity: $O(n_1 n_2 h_1 h_2)$, where n_1 and n_2 are number of nodes of the trees and h_1 and h_2 are height of the trees.