procedure Algorithm 2

input:
    generic compact representation of a neighborhood
    set $Q$ of query elements

output:
    comparison results of pertinent target sets with $Q$

begin
    Init queue with nodes corresponding to query elements
    while the queue is not empty do
        $T \gets$ next element of the queue
        $T$.tag $\gets$ pot_pert $\triangleright$ potentially pertinent

        $\triangleright$ step 1: test Rule 2 of pertinence definition
        if $\exists T' \in T$.children having $T$.common = $T'$.common then
            $T$.tag $\gets$ not_pertinent $\triangleright$ Rule 2 is violated

        $\triangleright$ step 2: test Rule 3 of pertinence definition
        for each $T' \in T$.children having $T'$.tag = pot_pert do
            if $T$.#differing = $T'$.#differing then
                $T'$.tag $\gets$ not_pertinent $\triangleright$ Rule 3 is violated
            else if $T$ is last parent processed of $T'$ then
                $\triangleright$ $T'$ is pertinent, performs comparison
                output(similarity_index($Q, T'$))

        if $T$ does not contain all the query elements then
            for each parent node $T'$ of $T$ do
                if $|T'| \leq$ max_target_size then
                    propagate common elements to $T'$
                    append $T'$ to the queue

    end while
end