```
function CALCULATEDELTA(Network G', Node n_0, Node n_1)
                  ▷ Assuming an edge (n_0, n_1) has just been added or removed from G'
    Δ⁻ ← MATRIX(NUMNODES(G), 73)      ▷ For storing the orbit counts of the removed graphlets
    Δ⁺ ← MATRIX(NUMNODES(G), 73)      ▷ For storing the orbit counts of the added graphlets
    B_0 = {n_0, n_1}                  ▷ A blacklist of nodes not to visit anymore
    e = (n_0, n_1)                    ▷ Different name for the edge
    if e ∈ N' then
        (m⁻, m⁺) = (0, 1)             ▷ In other words, m⁻ = 1 iff e ∈ N and m⁺ = 1 iff e ∈ N'
    else
        (m⁻, m⁺) = (1, 0)
    end if
    x_0 = 0
    (Δ⁺, Δ⁻) = COUNTORBITS(Δ⁺, Δ⁻, (n_0, n_1), x_0, m⁻, m⁺)   ▷ Update delta matrices for current nodes
    for n_2 ∈ ⋃_{i∈{0,1}} NEIGHBOURS(n_i) do
        if n_2 ∉ B_0 then
            B_0 = {n_2} ⋃ B_0             ▷ Add n_2 to blacklist B_0
            x_1 = x_0 + W(G', (n_0, n_2), 1) + W(G', (n_1, n_2), 2)   ▷ Calculate edge weights for current nodes
            (Δ⁺, Δ⁻) = COUNTORBITS(Δ⁺, Δ⁻, (n_0, n_1, n_2), x_1, m⁻, m⁺)
            B_1 = B_0                     ▷ Make a copy for the next iteration depth
            for n_3 ∈ ⋃_{i∈{0,1,2}} NEIGHBOURS(n_i) do
                if n_3 ∉ B_1 then
                    B_1 = {n_3} ⋃ B_1
                    x_2 = x_1 + W(G', (n_0, n_3), 3) + W(G', (n_1, n_3), 4) + W(G', (n_2, n_3), 5)
                    (Δ⁺, Δ⁻) = COUNTORBITS(Δ⁺, Δ⁻, (n_0, n_1, n_2, n_3), x_2, m⁻, m⁺)
                    B_2 = B_1
                    for n_4 ∈ ⋃_{i∈{0,1,2,3}} NEIGHBOURS(n_i) do
                        if n_4 ∉ B_2 then
                            B_2 = {n_4} ⋃ B_2
                            x_3 = x_2 + Σ_{i∈{0,1,2,3}} W(G', (n_i, n_4), i + 6)
                            (Δ⁺, Δ⁻) = COUNTORBITS(Δ⁺, Δ⁻, (n_0, n_1, n_2, n_3, n_4), x_3, m⁻, m⁺)
                        end if
                    end for
                end if
            end for
        end if
    end for
end function
function COUNTORBITS(Δ⁺, Δ⁻, Nodes S, Edgeweights x, Modifier m⁻, Modifier m⁺)
    L⁻ = L[x + m⁻]                ▷ Look up the orbits of the subgraph induced by S in N
    Δ⁻[S, L⁻] += 1                ▷ Increment orbit counts of nodes S at positions L⁻ in Δ⁻
    L⁺ = L[x + m⁺]                ▷ Look up the orbits of the subgraph induced by S in N'
    Δ⁺[S, L⁺] += 1                ▷ Increment orbit counts of nodes S at positions L⁺ in Δ⁺
    return (Δ⁻, Δ⁺)
end function
function W(Network G, Edge e, Exponent i)
    return e ∈ EDGES(G) ? 2^i : 0     ▷ Return 2^i if G contains edge e
end function
```