1. Partial Weighted Count Tree Construction Algorithm

Algorithm 1 Partial Weighted Count Tree()
Input: Database containing transactions TD
Output: Partial Weighted Count Tree: a compact and complete tree data structure

```
for each transaction T_r ∈ TD do
    for each item y ∈ T_r do
        y ← prime(y)
    end for each
    W_k ← product(T_k)
    if tree = empty then
        Nnew ← createNode(W_k)
    else
        node ← root.child
        done ← 0
        while done ≠ 1 do
            if weight(node) = W_k then
                count(node) ← count(node) + 1
                done ← 1
            else if W_k% weight(node)=0 then
                if node has no child then
                    Nnew ← createNode(W_k)
                    remove all factors in W_k that are in common with node
                    node.child ← Nnew
                    done ← 1
                else
                    remove all factors in node that are in common with W_k
                    assign node to the child of node
                end if
            else if weight(node)% W_k then
                Nnew ← createNode(W_k)
                place Nnew in place of node
                assign node as the child of Nnew
                remove all factors in node that are in common with Nnew
                done ← 1
            else
                assign node to the right of node
            end if
        end while
    end if
```

36: end for each