Algorithm 2 Rebuilding NCBI-Tree Denovo with Custom Taxonomy

Input: Custom taxonomy table with capacity $\mathcal{C}$ with column taxa as List $\mathcal{L}_{taxa7}$
Output: NCBI-TREE nodes.dmp and names.dmp for custom taxonomy

Initializing NCBI-Tree as Directed Acyclic Graph(DAG) with Dict $\mathcal{D}_{dag}$
Initializing vertices and edges in DAG with List $\mathcal{L}_{vertices}$ and $\mathcal{L}_{edges}$
Initializing taxa node ever seen with Dict $\mathcal{D}_{taxa_{seen}}$
Initializing taxa node ID with Dict $\mathcal{D}_{taxid_{node}}$

for $genome = 1, \mathcal{C}$ **do**
  for $\varepsilon, taxon$ in enumerate($\mathcal{L}_{taxa7}$) **do**
    Adding Vertex $taxon$ into $\mathcal{D}_{dag}$ when not in $\mathcal{D}_{dag}$, or nothing to do
    if $taxon$ NOT in $\mathcal{D}_{dag}$ **then**
      $\mathcal{D}_{dag}[taxon] = []$
      Updating Taxa Node ID with occurence order of incrementing by 1
      $\mathcal{D}_{taxid_{node}}[taxon] = \text{len}(\mathcal{D}_{taxid_{node}}) + 1$
    end if

    Updating Edges from root to leaf(RTL) in Linked List
    if $\varepsilon = 0$ **then**
      Adding Edge $taxon$ from root of $\mathcal{D}_{dag}$
      $\mathcal{D}_{dag}[root] = [taxon]$
    else
      Adding Edge $taxon$ to $taxon_{before}$ denoted with $\mathcal{L}_{taxa7}[\varepsilon - 1]$ of $\mathcal{D}_{dag}$
      $\mathcal{D}_{dag}[taxon_{before}] = [taxon]$
    end if
  end for
end for

Recursive DFS to generate node.dmp and names.dmp
Initializing nodes and names as List $\mathcal{L}_{name}$ and $\mathcal{L}_{node}$
DFS($taxon$, $\mathcal{L}_{name}$, $\mathcal{L}_{node}$)
for $taxon_{child}$ in $\mathcal{D}_{dag}[taxon]$ **do**
  if $taxon_{child}$ in $\mathcal{D}_{taxa_{seen}}$ **then**
    Continue
  end if
  $\mathcal{D}_{taxa_{seen}}[taxon_{child}] = flag$

  Appending nodes with $taxid, parent_id, rank, ...$ format
  $\mathcal{L}_{node}$.append($\mathcal{D}_{taxid_{node}}[taxon_{child}]$,$\mathcal{D}_{taxid_{node}}[taxon]$,$\mathcal{D}_{rank}[taxon_{child}]$,...)

  Appending names with $taxid, name, 'scientificname'$ format
  $\mathcal{L}_{name}$.append($\mathcal{D}_{taxid_{node}}[taxon_{child}]$,$taxon_{child}$,'scientific name')
  DFS($taxon_{child}$, $\mathcal{L}_{name}$, $\mathcal{L}_{node}$)
end for