Algorithm 1: Handling Inconsistent Blockchain Views (G)

Input: Received Blockchain graph G
Output: Authoritative view A_v
// Collect all valid sibling groups to S
$S \leftarrow \varnothing$
foreach $B \in G$ **do**
  $s \leftarrow \textsc{Sibling}(B)$
  $S.\textsc{Append}(s)$
end
// Initialize weights of each sibling groups
foreach $s \in S$ **do**
  $W(s) \leftarrow 0$
end
// Calculate weights of all sibling groups
foreach $B \in G$ **do**
  foreach $s \in S$ **do**
    foreach $v \in \textsc{SubviewsBeginWith}(s)$ **do**
      if $B \in v$ **then**
        $W(s) \leftarrow W(s) + \textsc{Miningpower}(B)$
        break
      end
    end
  end
end
// Find the authoritative view
$s \leftarrow \textsc{Siblings}(B^0)$ // B^0 stands for genesis block
$A_v \leftarrow \varnothing$
$A_v.\textsc{Append}(s)$
if $\textsc{ChildSiblingGroups}(s) = \varnothing$ **then**
  return s
else
  Update $s \leftarrow \underset{s' \in \textsc{ChildSiblingGroups}(s)}{\arg\max} W(s')$
  Go to line 21
end

Function: Subviews Begin with

```
   // Function to find all subviews that begin with s
 1 Function SubviewsBeginWith(s):
 2 |  V ← ∅
 3 |  if  ChildSiblingGroups(s) = ∅ then
 4 |  |  v ← ConstructSubview(s)
 5 |  |  V ← V.Append(v)
 6 |  |  return V
 7 |  else
 8 |  |  foreach c ∈ ChildSiblingGroups(s) do
 9 |  |  |  foreach v ∈ SubviewsBeginWith(c) do
10 |  |  |  |  V.Append(v.Append(s))
11 |  |  |  end
12 |  |  end
13 |  end
14 return V
```

Function: Child Sibling Groups

```
   // Function to find child siblings
 1 Function  ChildSiblingGroups(s):
 2 |  C ← ∅
 3 |  foreach s′ ∈ S do
 4 |  |  foreach B′ ∈ s′ do
 5 |  |  |  foreach B ∈ s do
 6 |  |  |  |  if B ∈ B′.predecessors then
 7 |  |  |  |  |  C.Append(s′)
 8 |  |  |  |  end
 9 |  |  |  end
10 |  |  end
11 |  end
12 return C
```

Algorithm 2: 1