Algorithm Topological sort using Kahn's algorithm [1]

```
procedure TOPOLOGICAL SORT()
    Sorted ← Empty list that will contain the sorted elements
    Queue ← Queue of all nodes with no incoming edge
    InDegree ← InDegree[n] = incoming edges of the node n

    for all node n in the graph do
        for all node m with an edge e from n to m do
            InDegree[m] = InDegree[m] + 1        ▷ number of incoming edges of the node m

    for all node n in the graph do
        if InDegree[n] is 0 then
            add n to Queue

    while Queue is non empty do
        remove a node n from Queue
        add n to tail of Sorted
        for all node m with an edge e from n to m do
            InDegree[m] = InDegree[m] − 1
            if InDegree[m] is 0 then        ▷ m has no other incoming edges
                insert m into Queue

    if grap has no edges then
        return error        ▷ graph has at least one cycle
    else
        return Sorted        ▷ a topologically sorted order
```

Example (Where S = Queue and L = sorted):

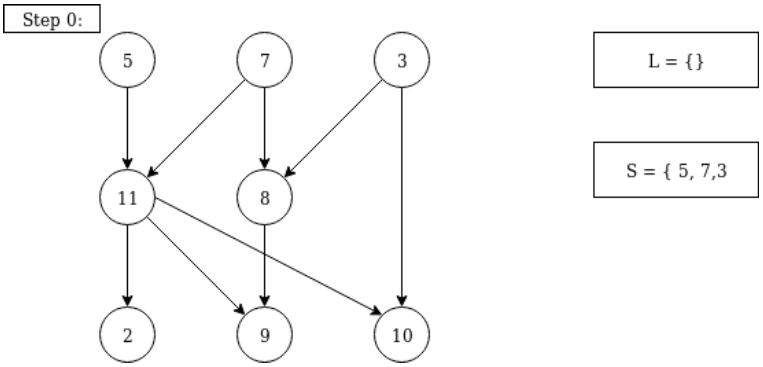

Figure 1: init state

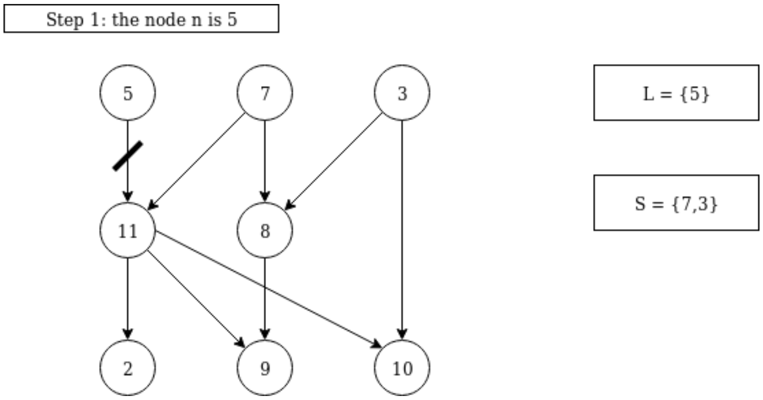

Figure 2: state 1

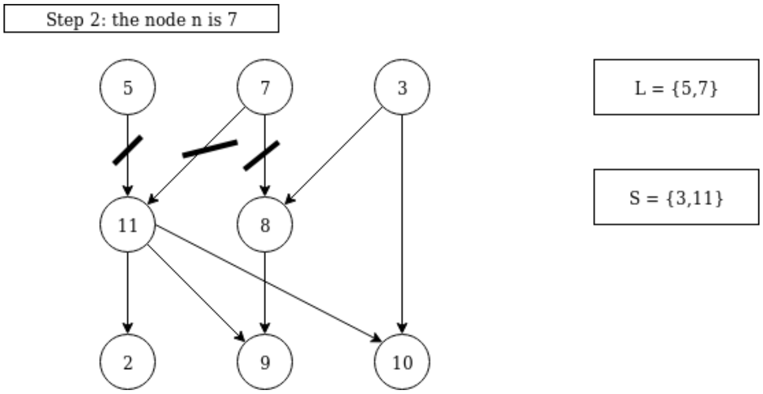

Figure 3: state 2

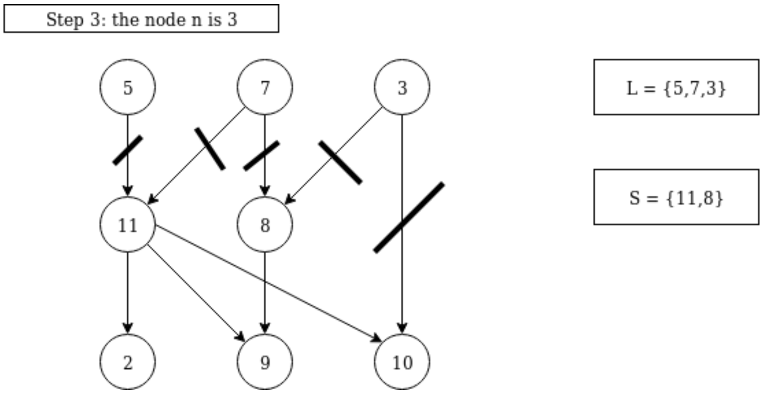

Figure 4: state 3

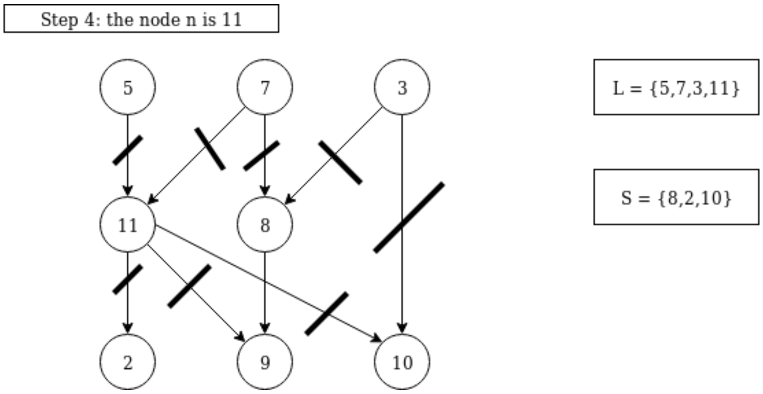

Figure 5: state 4

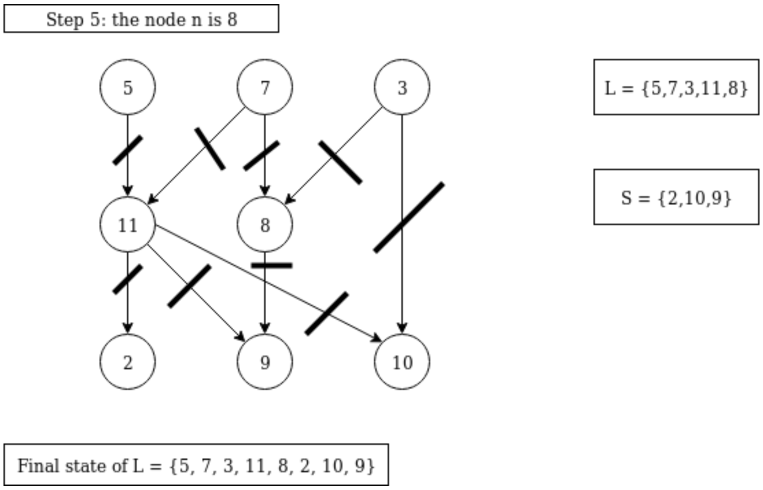

Figure 6: state 5

References

[1] Topological sort: Kahn's algorithm
https : //en.wikipedia.org/wiki/Topological_sorting#Kahn's_algorithm