## Acyclic Test Algorithm

**Input:** a directed graph $G(V, E)$
**Ouput:** **true** if $G$ is acyclic and **false** otherwise

1. $ZERO \leftarrow \{\ v \in V \mid indeg(v) = 0\ \}$
2. **while** $ZERO \neq \emptyset$
3. **do** choose arbitrary $v$ from $ZERO$;
4. delete $v$ from $ZERO$;
5. **forall** $w$ with $(v, w) \in E$
6. **do** delete $(v, w)$ from $G$;
7. **if** $indeg(w) == 0$
8. **then** add $w$ to $ZERO$; **fi**
9. **od**
11. **od**
12. **return** $|E| == 0$;

```
bool ACYCLIC(graph G)
{
  list<node> ZERO;
  node u;
  forall_nodes(u,G)
    if (G.indeg(u) == 0) ZERO.append(u);

  while (!ZERO.empty())
  { node v = ZERO.pop();
    forall_adj_edges(e,v)
    { node w = G.target(e);
      G.del_edge(e);
      if (G.indeg(w) == 0) ZERO.append(w);
     }
   }
  return G.number_of_edges() == 0;
}
```