Ryan Young

CSE 3100

9/24/2019

DAG's and Strong Components

1.) Prove the following statement:
In every directed acyclic graph, there is a node with no outgoing edges.

Proof:

Suppose that G is a graph where every node has at least on outgoing edge.
If we can show that we can find a cycle, this proves the claim.
Pick any node v and follow the edges forwards.
This is possible since all nodes have at least one outgoing edge.
Take n + 1 steps.
Now, at least one node has been visited twice thus proving G has a cycle.

2.) Psuedo Code:

Choose starting point (u)

Check if path from u to v and v to u // checking if u and v are strong components

If so append v as a strong component in a list based of starting point u

Else:

Move to new neighbor and run algorithm again on that node

The computational complexity of is actually the same as BFS (Bredth-First Search) giving it a time complexity of O(V+E) where V is the number of vertices and E is the number of edges.

3.) Prove the following statement:
The meta-graph over the strong components of a directed graph is a directed acyclic graph.

Proof:

Suppose that G is a directed graph.
By definition of strong components, if u is a strong group and v is another strong group then it is impossible to go from u to v and from v to u.
Since it is impossible to travel in both directions between the groups this means the graph is directed.
By definition of a directed graph all nodes can only have either incoming our outgoing edges.
This infers at least one node does not have an incoming edge.
By statement 3.19, in a DAG there should be a node with no incoming edges, we know that this is a meta graph is a Directed Acyclic Graph.