Justin Cabral

CS 5084

Assignment 3

Graphs

Graphical user interface

Description automatically generated with low confidence

The graph in figure 3.10 shows that node ‘a’ must come first because it has no edge coming into it, and ‘f’ must come last because there is no edge leaving it. For our remaining four nodes, b must precede c, and d has to precede e. Other than that, we can place the remaining 4 nodes however we feel necessary.

Knowing this we can form the following topological orderings:

* a,b,c,d,e,f
* a,b,d,e,c,f
* a,b,d,c,e,f
* a,d,b,c,e,f
* a,d,b,e,c,f
* a,d,e,b,c,f

Therefore, graph G has 6 topological orderings.

Text

Description automatically generated

I will use a BFS algorithm seeing that the work performed will be constant for each ‘n’ node and ‘m’ edge, which results in a running time of O(m + n).

To run the BFS on a given undirected graph, we must start from node ‘s’ which will return us a tree T. If all edges of G happen to appear in the BFS tree, then we know G = T and G is a tree that contains no cycles, thus being acyclical. However, if there is an edge in our original graph (u,v) that is not in our BFS tree, it means that there is a path from the starting node ‘s’ to node ‘u’, and a path from node ‘s’ to node ‘v’. If we were to introduce the edge (u,v) into the BFS tree would create a cycle.

Text

Description automatically generated

Base:

* Let h = 0, where h is the height of a binary tree
* Given a tree of height 0, there is exactly 1 leaf which is the root
* Thus 20 – 1 = 1 – 1 = 0 given there are 0 nodes with 2 children
* If we add 2 children to the tree with exactly 1 node, our results are 20+1 – 1 = 21 -1 = 2-1 = 1
* Our hypothesis has been satisfied
* Assume that for h >= 0, where h is the height of a binary tree, there is at most 2h+1 – 1 nodes

Induction:

* Let T be a binary tree of height h + 1
* That means all nodes found at level h+1 are leaf nodes
* There are at most 2(h+1) leaves
* Removing all the leaves would leave us with a tree of height h
* Therefore, the total number of nodes remaining in tree T are 2(h+1) – 1
* Meaning the total number of nodes in T is at most 2(h+1) + 2(h+1) – 1 = 2(h+1+1) – 1
* Thus, we have proven our claim by proof of induction

Text

Description automatically generated

We can solve this by using proof by contradiction:

* Suppose G has an edge of e ={x,y} that doesn’t belong to tree T
* Given T is a depth-first search tree, one of the two ends must be an ancestor of the other
* If ‘x’ is an ancestor of ‘y’, the distance of the two nodes from ‘u’ in T can differ by at most 1
* However, if ‘x’ is an ancestor of ‘y’, then distance of ‘u’ to ‘y’ in tree T is at most always one greater than the distance from ‘u’ to ‘x’.
* That means ‘x’ must be the direct parent of ‘y’ in tree T
* This means {x,y} is an edge of tree T, which contradicts our initial assumption
* Thus, we have solved G = T using proof by contradiction

Text

Description automatically generated with medium confidence

Text

Description automatically generated

This claim is **true** which will be proved by contradiction:

* Let G be a graph the properties associated with in the questions claim
* Suppose by way of contradiction that it is not connected
* Let X be the nodes in the smallest connected component
* Knowing there are at least two connected components we know |X| <= n/2
* Consider that for any node v in X
* All neighbors must lie in X, so its degree is at most |X| - 1 <= n/2 – 1 < n/2
* This contradicts the questions original claim that every node has a degree of at least n/2
* Thus, we have proved the claim to be true by contradiction