Justin Cabral

CS 5084

Assignment 3

Graphs

Exercises

1. Consider the directed acyclic graph G in Figure 3.10. How many topological orderings does it have?

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.

2. Give an algorithm to detect whether a given undirected graph contains a cycle. If the graph contains a cycle, then your algorithm should output one. (It should not output all cycles in the graph, just one of them.) The running time of your algorithm should be $O(m+n)$ for a graph with n nodes and m edges.

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.

5. A binary tree is a rooted tree in which each node has at most two children. Show by induction that in any binary tree the number of nodes with two children is exactly one less than the number of leaves.

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 $2^0 - 1 = 1 - 1 = 0$ given there are 0 nodes with 2 children
- If we add 2 children to the tree with exactly 1 node, our results are $2^{0+1} - 1 = 2^1 - 1 = 2-1 = 1$
- Our hypothesis has been satisfied
- Assume that for h >= 0, where h is the height of a binary tree, there is at most $2^{h+1} - 1$ 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

6. We have a connected graph $G = (V, E)$, and a specific vertex $u \in V$. Suppose we compute a depth-first search tree rooted at u, and obtain a tree T that includes all nodes of G. Suppose we then compute a breadth-first search tree rooted at u, and obtain the same tree T. Prove that $G = T$. (In other words, if T is both a depth-first search tree and a breadth-first search tree rooted at u, then G cannot contain any edges that do not belong to T.)

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

7. Some friends of yours work on wireless networks, and they're currently studying the properties of a network of n mobile devices. As the devices move around (actually, as their human owners move around), they define a graph at any point in time as follows: there is a node representing each of the n devices, and there is an edge between device i and device j if the physical locations of i and j are no more than 500 meters apart. (If so, we say that i and j are "in range" of each other.)

They'd like it to be the case that the network of devices is connected at all times, and so they've constrained the motion of the devices to satisfy the following property: at all times, each device i is within 500 meters of at least $n/2$ of the other devices. (We'll assume n is an even number.) What they'd like to know is: Does this property by itself guarantee that the network will remain connected?

Here's a concrete way to formulate the question as a claim about graphs.

Claim: Let G be a graph on n nodes, where n is an even number. If every node of G has degree at least $n/2$, then G is connected.

Decide whether you think the claim is true or false, and give a proof of either the claim or its negation.

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