Diego Valdes

CSCE 350

Oct 10, 2016

Project 2

**Topological Sort**

I designed an algorithm that does DFS on a graph recursively. In DoTopoSort() I have a for loop that starts at 0 and traverses the entire graph checking if each node has been visited. If the node hasn’t been visited, a call is made to the recursive function MagicBaby().

MagicBaby() itakes an int parameter which is the index of the node which hasn’t been visited. It starts off by setting the node in the graph to visited. It then gets the vector of children nodes and iterates through it looking to see if those children have been visited. If they have been visited it iterates to the next child. If the child hasn’t been visited, it calls itself with the index of the child that hasn’t been visited. If a node has been visited, it is placed in a stack. The cycle continues until the entire graph has been visited.

Once MagicBaby() has finished its job, DoTopoSort() is responsible for printing the stack. This is achieved with a while loop and the top() and pop() class functions. The top is printed and then popped. Once the stack is empty, we now have a printed list of topologically sorted nodes.

I choose to do this project in the recursive way because now that I’m a Junior and have taken some other courses, I believe I have a better understanding of recursion and how it can be used. After building the algorithm initially, I went back and tweaked a few things for efficiency and I had some redundancies that weren’t needed.