Albert Chieu

CSC 220

7/28/11

Topological Sort Discussion

**What is the problem?**

Given a directed graph of a real world example of getting dressed, provide a complete ordering using a topological sort.

**What is the algorithm?**

The algorithm is a linear ordering of all vertices of a directed graph such that, for every edge, the source vertex comes before the target vertex with partial ordering.

**How does topological sort work?**

Use a Depth First Search to go through all the vertices and output the vertices in reverse topological order with the following steps:

1. List all nodes without any incoming edges and pick one node to put into a sorted array.
2. Remove that node and their outgoing edges from the graph.
3. Repeat Steps 1 and 2 until all nodes have been picked into the sorted array.

**How is the program implemented?**

Define both a node and edge classes. The node class holds the name and a list of incoming and outgoing edges. The edge class defines the source and target nodes.

The directed graph for the getting dressed example is defined with the given nodes and edges. All the input nodes are arranged into an array for the sorting algorithm specified in the previous section. The output is a sorted array with partial ordering with its size equal to the total number of nodes. If the output size is not equal to the input size, then there exists a cycle in the directed graph and the nodes cannot be topologically sorted.

**Generally how well does it do?**

This topological sorting algorithm is a greedy method that gives the first available solution. This is because there might be more than one element in the list of nodes with empty incoming edges. By picking a different element from this list, a different solution will be resulted.