***Graph Algorithms in MATLAB***

done by Shaik Ameer Basha, ME13B116, IITM

under the guidance of Prof. Rupesh Nasre

First algorithm - Dominators

Dominator: In a directed graph, for a given source and target nodes, a node which occurs in all paths from the given source node to the given target node, is called a dominator.

- So, by definition, given *target node* will always be a dominator. It is called self-dominator.

- Dominator which isn't self is called a ***strict dominator***.

Code:

*Input*: 1) A text file containing source nodes matrix and target nodes matrix as two columns.

- Text file can also have description of the data present. If there are just two columns and no description, then comment lines (8 and 9) and uncomment lines (11, 12, 13 and 14).This would save some time for large graphs.

2) Source and target nodes

*Output*: 1) Shortest path from the given source node to the given target node.

2) Possible strict dominators.

Algorithm:

1) Find shortest path between given source and target nodes. Because dominator(s) must occur in all paths, it would just be enough if nodes in this shortest path are checked for their occurrence in other paths.

2) Consider given source node.

* Get the end nodes of edges in which this source node is the start node.
* Out of all these, take only those nodes, which further lead to given target node without coming back to source node.
* Find all shortest paths from these nodes to the target node. One among these will be initially found shortest path.
* Compare them all and find nodes which are common in all, which is the output here.

3) Record first element of the above output in a matrix (D1) and repeat step 2, for this first element. Again record first element of the output and proceed the same way until target node is the output.

4) Now, given source nodes matrix and target nodes matrix are interchanged. Also source and target nodes are also interchanged.

5) Repeat steps 1, 2 and 3 and obtain another matrix (D2) containing first elements of outputs from step 2.

6) Compare these two matrices (D1 and D2), to get set of ***possible strict dominators***.

Remarks:

Because above method doesn't validate the definition of dominator, it will only give a possible answer. To find the exact answer, we need all paths from given source to destination node, for which there isn’t a way as of now.

Results are in [Results.xlsx, Sheet 1](E:\\Docs\\mine\\graphing techniques project\\project_final\\Results.xlsx).

Second algorithm – Graph coloring

*Vertex coloring:* To assign colors to nodes (vertices) in a graph such that no two adjacent nodes (vertices) have the same color.

Code:

*Input*: A text file containing source nodes matrix and target nodes matrix as two columns.

- Text file can also have description of the data present. If there are just two columns and no description, then comment lines (8 and 9) and uncomment lines (11, 12, 13 and 14). This would save some time for large graphs.

*Output*: Matrix containing RGB values for colors of nodes.

Algorithm: used is **Greedy coloring algorithm**.

Greedy coloring algorithm:

1) Color first vertex with first color.

2) Do following for remaining vertices.

- Consider the currently picked vertex and color it with the lowest numbered color that has not been used on any previously colored vertices adjacent to it. If all previously used colors appear on vertices adjacent to v, assign a new color to it.

Remarks:

* MATLAB automatically adds missing nodes in between. While assigning colors, these are not taken into account.
* Plotting the nodes and coloring is taking up so much time. So just the color values are noted in a matrix.
* Results are in [Results.xlsx, Sheet 2](file:///E:\Docs\mine\graphing%20techniques%20project\project_final\Results.xlsx).