Documentation Graphs

This documentation is made by 214101020 Harsh Bijwe.
There are 2 folders and 1 cpp file zipped together.

1. *input folder*: Contains 10 pre-made Test-files which is used as input source to 214101020_Graphs.cpp. It will also contain a custom made test case.
2. *Graphviz folder*: It will contain Output of each of the 4 Operations of Assignment, along with originalGraph output.
3. *Cpp File*: Main file where all computations happen.

NOTE: It is Mandatory to place above mentioned files & folders in the same Directory. Otherwise, it will lead to segment fault. Other details on how to run can be found in the *Readme file* which is zipped along.

Logic Used:

1. DFS : DFS is implemented as a recursive procedure. We track *visited node vector* instead of coloring each vertex. *Pre* & *post* vectors hold the start and end time of each node as a vector.

2. Tarjan SCC: We maintain a *low number* and *df number* per node as a vector to find out SCC. This method implicitly uses DFS. *Low number* is the smallest *df numbered vertex* that can be reached from a given vertex by using one or more tree edges and **at most**

one back-edge or cross-edge. *Df number* is the depth first number of the node.

3. Min-Edge Graph: First find SCC using Tarjan's Algorithm. As per the edges in the original graph, find out edges within SCC & intra-SCC. Also reject parallel edges.
4. Is Semiconnected: It can be done in O(V*V*V) complexity [Floyd-Warshall Algorithm] but an efficient Algorithm exist whose time complexity is O(V+E):-
    a. Find Maximal SCCs in the graph
    b. Build the SCC graph G'=(U,E') such that U is a set of SCCs. E'= {(V1,V2) | there is v1 in V1 and v2 in V2 such that (v1,v2) is in E)
    c. Do topological sort on G'
    d. Check if for every i, there is edge Vi,Vi+1.
5. Dijkstra: Dijkstra's algorithm uses a priority queue implemented as Min-heap Data Structure, which gives the least reachable node from source to target. Each time we find a vertex, we close it. And repeat the procedure. It is NOT guaranteed that dijkstra computes the shortest path in case of negative weight edges & negative Cycles, it may fail to give the shortest path in these 2 cases.