HW4: Well-Connected Graph

Describe your algorithm in words:

Step 1:

We have the graph G = (V, E) and run the SCC algorithm on G.

We can get back: G_SCC = (V_SCC, E_SCC), where each SCC is condensed to a single vertex and edges are added between vertices based on the origin edges in G.

Step 2:

The G_SCC is s DAG, we apply DFS to get topological sort on G_SCC.

We can get back a sorted sequence of vertices of G_SCC: {v[1], v[2], … , v[n]}

Step3:

For the sorted vertices, if there exists an edge between each pair of neighboring vertices, the given graph is well-connected. Otherwise, it is not well-connected.

Justify its correctness :

1) After applying the SCC algorithm, all vertices within a SCC could reach all other vertices in the same SCC.
2) In a DAG, if there exists an edge between each pair of neighboring vertices after the topological sort, it means that any of these vertices can reach any vertex behind it. Thus, all vertices within an SCC[p] could reach all vertices in the SCC[q] (SCC[q] are behind the SCC[p] after topological sort).
3) Therefore, if the condition is satisfied, for every pair of distinct vertices u and v in V, if they are in the same SCC, they could reach each other; if they are in different SCCs, there exists a way between their located SCC, and that means there exists a path between them, so it could be concluded that the given graph is well connected.

Analyze and state its running time. :

Given n = | V | and m = | E |.

SCC algorithm takes O(n +m), applying DFS to get topological sort on G_SCC and determining the presence or absence of an edge between each pair of neighboring vertices take O(n+m), Thus, the overall running time is O(n + m).