**Find if there is a path between two vertices in a directed graph**

We can either use Breadth First Search (BFS) or Depth First Search (DFS) to find path between two vertices. Take the first vertex as source in BFS (or DFS), follow the standard BFS (or DFS). If we see the second vertex in our traversal, then return true. Else return false.

**Strongly Connected Graph:**

A directed graph is strongly connected if there is a path between any two pair of vertices.

**How To Check If A Graph Is Strongly Connected Or Not In Case Of Undirected Graph:**

It is easy for undirected graph, we can just do a BFS and DFS starting from any vertex. If BFS or DFS visits all vertices, then the given undirected graph is connected. This approach won’t work for a directed graph. For example, consider the following graph which is not strongly connected. If we start DFS (or BFS) from vertex 0, we can reach all vertices, but if we start from any other vertex, we cannot reach all vertices.

**How To Check If A Graph Is Strongly Connected Or Not In Case Of Directed Graph:**

**A simple idea** is to use a all pair shortest path algorithm like Floyd Warshall or find Transitive Closure of graph. Time complexity of this method would be O(v3).

**Another idea** is doing BFS or DFS for all vertices. If any DFS, doesn’t visit all vertices, then graph is not strongly connected. This algorithm takes O(V\*(V+E)) time which can be same as transitive closure for a dense graph.

**Far Better idea:**

1) Initialize all vertices as not visited.

2) Do a DFS traversal of graph starting from any arbitrary vertex v. If DFS traversal doesn’t visit all vertices, then return false.

3) Reverse all arcs (or find transpose or reverse of graph)

4) Mark all vertices as not-visited in reversed graph.

5) Do a DFS traversal of reversed graph starting from same vertex v (Same as step 2). If DFS traversal doesn’t visit all vertices, then return false. Otherwise return true.

The idea is, if every node can be reached from a vertex v, and every node can reach v, then the graph is strongly connected. In step 2, we check if all vertices are reachable from v. In step 4, we check if all vertices can reach v (In reversed graph, if all vertices are reachable from v, then all vertices can reach v in original graph)

**Articulation Points In A graph:**

A vertex in an undirected connected graph is an articulation point (or cut vertex) iff removing it (and edges through it) disconnects the graph. Articulation points represent vulnerabilities in a connected network – single points whose failure would split the network into 2 or more disconnected components. They are useful for designing reliable networks.

For a disconnected undirected graph, an articulation point is a vertex removing which increases number of connected components.