Module 2 Graphs Written Assignment

Tyler Kim
tkj9ep

October 05, 2022

1 Responses

1. Let the adjacency matrix, Adj, for G = (V, E), represent the edges of G such that Adj[u][v] denotes an edge from node u to node v. Let an array, A, of size 1 x V denote whether a particular node could possibly be a sink node and be initialized to all 1's. In other words, if the value of A[x] is 1, then node x could possibly be a sink node, otherwise node x is not. Let a pointer, ptr, point to the first element of A.

The algorithm would start at Adj[u][v] where u, v = 0. If the algorithm finds the value of 0 at Adj[u][v] and u != v, then A[v] will be set to 0 and will move one index to the right (v = v + 1). Otherwise, if the algorithm finds Adj[u][v] = 1, then A[u] is set to 0 and the algorithm will look one index down (u = u + 1). If ptr is not pointing to a value of 1 in A, ptr will traverse through A until it reaches an index with the value of 1. Once the algorithm reaches the end of Adj, it will check if the last node is the sink node by seeing if its row is all 0's and its column is all 1's except where the edge is to itself.

2.

```
DFS(G)
    for each vertex u in G.V
        u.color = WHITE
    for each vertex u in G.V
        if u.color == WHITE
            DFS-VISIT(G, u)

DFS-VISIT(G, u)
    u.color = GRAY
```

```
for each v in G.Adj[u]

    if v.color == WHITE // recursive case
        DFS-VISIT(G, v)
    u.color = BLACK
```

3.

4.