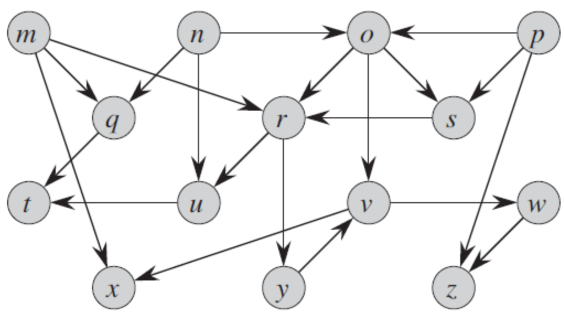
Homework 8

2019.04.25

Note: When the exercise asks you to "design an algorithm for…," it always means that "designs an EFFICIENT algorithm for … and ANALYZES your algorithm". You should keep this in mind when writing solutions.

1. Exercises 22.4-2

Give a linear-time algorithm that takes as input a directed acyclic graph $G = (V, E)$ and two vertices s and t, and returns the number of simple paths from s to t in G. For example, the directed acyclic graph of following figure contains exactly four simple paths from vertex p to vertex v: *pov*, *poryv*, *posryv*, and *psryv*. (Your algorithm needs only to count the simple paths, not list them.)



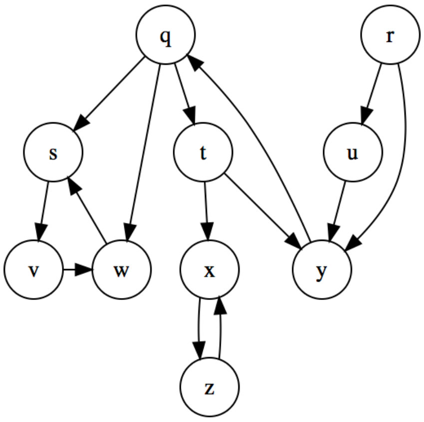
2. Exercises 22.4-3

Give an algorithm that determines whether or not a given undirected graph G = (V, E) contains a cycle. Your algorithm should run in O(V) time, independent of |E|.

3. A DFS forest can be generated by perform DFS on a directed graph. There are 4 types of edges in a DFS forest: tree edge, forward edge, back edge and cross edge. Modify DFS so that it can determine the type of each edge.

4. Exercises 22.5-2

Show how the procedure STRONGLY-CONNECTED-COMPONENTS works on the following graph. Specifically, show the finishing times computed in line 1 and the forest produced in line 3. Assume that the loop of lines 5-7 of DFS considers vertices in alphabetical order and that the adjacency lists are in alphabetical order.




```
DFS(G)
for each vertex u ∈ G.V
    u.color = WHITE
    u.π = NIL
time = 0
for each vertex u ∈ G.V
    if u.color == WHITE
        DFS-VISIT(G, u)
```

```
DFS-VISIT(G, u)
time = time + 1              // white vertex u has just been discovered
u.d = time
u.color = GRAY
for each v ∈ G.Adj[u]        // explore edge (u, v)
    if v.color == WHITE
        v.π = u
        DFS-VISIT(G, v)
u.color = BLACK              // blacken u; it is finished
time = time + 1
u.f = time
```

```
STRONGLY-CONNECTED-COMPONENTS(G)
1  call DFS(G) to compute finishing times u.f for each vertex u
2  compute G^T
3  call DFS(G^T), but in the main loop of DFS, consider the vertices
       in order of decreasing u.f (as computed in line 1)
4  output the vertices of each tree in the depth-first forest formed in line 3 as a
       separate strongly connected component
```

5. Exercises 22.5-7

A directed graph G = (V, E) is **semiconnected** if, for all pairs of vertices u, v ∈ V, we have u ~> v or v ~> u. Give an efficient algorithm to determine whether or not G is semiconnected. Prove that your algorithm is correct, and analyze its running time.

6. EXT. 7-2

Slide Unit7 P. 39 有提到另一個 DFS 的實作，但其輸出的 order 可能會不同，試問該 DFS 的實作是否可以用來解 Topological Sorting？ 若可以，請將該 DFS 的實作改成可以解 Topological Sorting。

7. EXT. 7-5

Let G = (V, E) be a connected undirected graph. Give the sufficient and necessary condition that the edges of G can **be oriented** such that the resulting directed graph is **strongly connected** and design the algorithm to **find such an orientation if it exists**. (Hint: "be oriented" means adding directions to the edges of the undirected graph.)