7.Z

a)

```
// driver and preprocessing
global G = (V,E);
global Adj[1...n];
global root[];
procedure driver(Adj[1...n]);
   for k in Adj[1...n] do
      if k is null then
         // k is a leaf
         k.longestoutof <- 0;
         mark k as done;
      else
         mark k as undone;
      end if;
   end for;

   if r in Adj[1...n] is not in other vertices' Adj then
      // r is a root
      root.append(r);
   end if;

   while root is not empty do
      m <- root.pop();
      longest(m);
   end while;
end_driver;


// DFS

procedure longest(vertex m);
   if m is undone then
      for k in Adj[m] do
         m.longestoutof <- max(k.longestoutof) + Ecost(m, k);
      end for;
      mark m as done;
   end if;
end_longest;
```

Correct

b)

```
global G = (V,E);
global Adj[1...n];

// prepare all nodes
mark all nodes as not done and not started;
mark roots as done;
mark all roots' longestinto as 0;
for roots x do
   TarjanDFS(G,x);
end for;

procedure TarjanDFS(G;;w);
   mark w as started;
   for all neighbors m of w do
      if m is unstarted then
         m.longestinto <- max(w.longestinto + Ecost(w,m), m.longestinto)
         TarjanDFS(m);
      else if m is undone then // there is a cycle
         start cycle processing;
      end if;
   end for;
   print(w);
   mark w as done;
end_TarjanDFS;
```

Correct

c)

Use part b's answer as a's input.

Incorrect. I did not think of reversing the edges.

d)

Use part a's answer as b's input.

Incorrect. I did not think of reversing the edges.