Problem 8.4

Longest Path

```
Require: a DAG $(V, E)$ and a source node $s$
Ensure: a $distance$ array, stores the longest distance to each vertex $v$ from $s$
  function SSLP($V, E, s$)
     for $v \in V$ do
        $distance[v] \leftarrow -\infty$
     end for
     $distance[s] \leftarrow 0$
     $V^* = V$, sorted in topological order
     for $v \in V^*$ do
        for $u|(u, v) \in E$ do
           $distance[v] \leftarrow max\{distance[v], distance[u] + w(u, v)\}$
        end for
     end for
  end function
```

Longest Path Dijkstra's

```
Require: a graph $(V, E)$ and a source node $s$
Ensure: a $distance$ array, stores the longest distance to each vertex $v$ from $s$
  function SSLP($V, E, s$)
    $PQ$ = max-heap priority queue
    for $v \in V \backslash \{s\}$ do
      $distance[v] \leftarrow -\infty$
      $PQ$.INSERT($v, distance[v]$)
    end for
    $distance[s] \leftarrow 0$
    $PQ$.INSERT($s, distance[s]$)
    while $\neg PQ$.ISEMPTY() do
      $u \leftarrow PQ$.DELETEMAX()
      for $u|(u, v) \in E$ do
        if $distance[v] < distance[u] + w(u, v)$ then
          $distance[v] \leftarrow distance[u] + w(u, v)$
          $parent[v] \leftarrow u$
          $PQ$.INCREASEKEY($v, distance[v]$)
        end if
      end for
    end while
  end function
```

The above algorithm does not give the longest paths in a cyclic graph. Consider the graph in fig. 1.

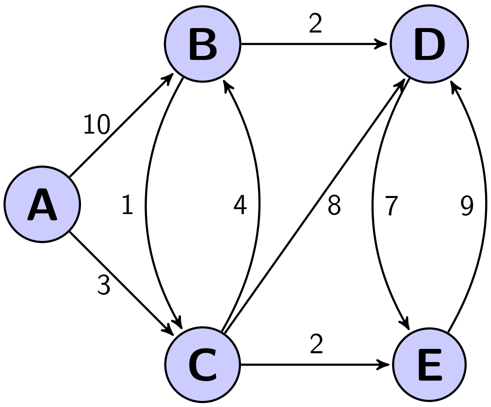

fig 1.

Following this algorithm would result in the following steps:

A	B	C	D	E	PQ
0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$A_0 B_{-\infty} C_{-\infty} D_{-\infty} E_{-\infty}$
0	10	3	$-\infty$	$-\infty$	$\cancel{A_0} B_{-\infty} C_{-\infty} D_{-\infty} E_{-\infty}$
0	10	11	12	$-\infty$	$\cancel{B_{10}} C_3 D_{-\infty} E_{-\infty}$
0	10	11	12	19	$\cancel{D_{12}} C_{11} E_{-\infty}$
0	10	11	21	19	$\cancel{E_{19}} C_{11}$
0	10	11	21	19	$\cancel{C_{11}}$

The values in the distance array in the bottom row when the algorithm terminates are not the longest paths. For instance, the longest distance to D is 22, along the path A-B-C-E-D. The other problem is that the algorithm does not find a simple path. The distance found in the array for D, 21, is from the path A-B-D-E-D, which passes through D twice.