Homework 9, additional problems, solution set

1. Let $e = (u, v)$ and let $G^e$ denote the graph $G$ with edge $e$ removed. We observe that if $G$ does not have a cycle containing $e$ then the component of $G$ that previously contained $e$ will become two components in $G$. For otherwise, there is a path connecting the endpoints $u$ and $v$ of $e$, and adding $e$ to this path forms a cycle contained $e$. Conversely, if there is a cycle containing $e$ in $G$, then removing $e$ leaves $u$ and $v$ connected, so the component in $G$ that contained $e$ is still a component in $G^e$.

This leads to the following algorithm. Perform a DFS on $G^e$ that numbers the vertices with their component numbers. Let $e = (u, v)$. If $u$ and $v$ have distinct numbers then $G$ does not have a cycle containing $e$ and otherwise it does. As DFS runs in linear time, so does this algorithm.

2.a. In general graphs, cycles of negative length could create paths of arbitrary negative length (by going around such a cycle repeatedly). This issue does not arise in dags. In a dag, a shortest path from $v$ will go to a neighbor of $v$, and from there follow a shortest path to the destination. This leads to the following algorithm.

We perform a DFS on $G$. For each $v \in V$ we will determine the length of a shortest path to $t$, storing the results in the array Shtst$[1 : n]$, by augmenting the DFS as follows. We take the minimum of the following value:

$$\text{Shtst}[v] \leftarrow \begin{cases} \min_{(v,w)\in E}\{\text{length}(v, w) + \text{Shtst}[w]\} & \text{if } v \neq t \\ 0 & \text{if } v = t \end{cases}$$

To compute this we initialize Shtst$[v]$ to maxint for all $v \neq s$, and Shtst$[t]$ to 0. Pseudo-code for the recursive computation follows.

```
DFS-Sht(v)
Done[v] ← True;
for each edge (v, w) do
    if not Done(w) then DFS-Sht(w)
    end if
    Shtst[v] ← min{Shtst[v], length(v, w) + Shtst[w]}
end for
```

b. We perform the same computation on $G^R$, the reversal of $G$, with $s$ as the target vertex. For a path from $v$ to $s$ in $G^R$ is the reversal of a path from $s$ to $v$ in $G$.

3. The answer is "no". Suppose $G$ has 3 vertices, $u, v, w$, and edges $(u, w)$ and $(v, w)$. Suppse the search starts at vertex $w$, then goes to $u$ and finally to $v$. Then post$(u) = 1$ and post$(v) = 2$. But $u$ and $v$ are not ancestors of each other (they are both children of $w$).