Homework 10, additional problems, solution set

1. We compute the strong components in G, forming the meta-graph H of strong components, which is a dag. As we do this, we also compute the sum of the weights in each strong component, and these sums become the weights for the corresponding nodes in the meta-graph.

Recall that in the second DFS of G the DFS explores the strong components one by one. For each strong component, as each vertex is explored, we increment a running total of the weight of the vertices explored. The final value of this sum will be the weight of the nodes in that strong component.

This increases the run-time of the DFS by $O(n)$ so this remains a linear time computation.

We now run the given algorithm on the meta-graph. The final step is to propagate the sum from each meta-graph node to each vertex in the corresponding strong component.

The computation of the strong components determines a component number for each vertex, and these numbers are also used to number the vertices in the meta-graph H. After the sum of values algorithm is run on H, each of its vertices x has the value of the reachable weight stored in array entry $\mathrm{ValG}[x]$ say, where x is the component number for the corresponding component. Then the calculation $\mathrm{ReachWt}[v] \leftarrow \mathrm{ValG}[v.\mathit{cpt}]$ for each vertex $v \in G$ determines the desired sums of weights. This takes a further $O(n)$ time.

2. We compute the height of a path $P \circ e$ as follows:

$$\mathrm{height}(P \circ e) = \max\{\mathrm{height}(P), \mathrm{height}(e)\}.$$

This changes the inner loop of Dijkstra's algorithm to become:

for each edge (u, v) **do**
    if $\mathrm{Dist}[v] > \max\{\mathrm{Dist}[u], \mathrm{height}(u, v)\}\}$ **then**
        $\mathrm{Dist}[v] \leftarrow \max\{\mathrm{Dist}[u], \mathrm{height}(u, v)\}\}$
        $\mathrm{DecreaseKey}(Q, v, \mathrm{Dist}[v])$
    end if
end for

The correctness argument is unchanged as it just relies on the property that $\mathrm{length}(P \circ e) \geq \mathrm{length}(P)$, which in the current context means we need $\mathrm{height}(P \circ e) \geq \mathrm{height}(P)$, which is indeed the case.

The runtime of the modified algorithm is the same as for the standard Dijkstra's algorithm as each iteration of this loop still takes $O(1)$ time plus the time for a DecreaseKey operation.