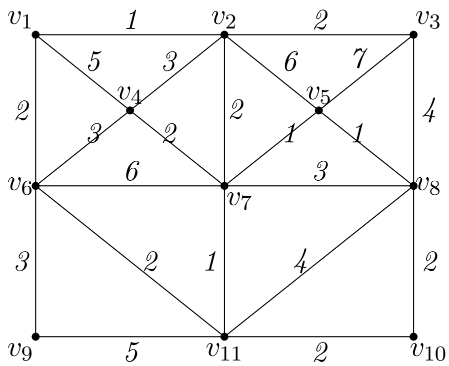
MAS210 Graph Theory Exercises 4 Solutions

Q1 *Consider the following network N.*

$N =$ 

An implementation of Dijkstra's algorithm starting at v_1 produces the following tree T_4 at the end of the fourth iteration: $V(T_4) = \{v_1, v_2, v_3, v_6\}$ and $E(T_4) = \{v_1v_2, v_2v_3, v_1v_6\}$. It also gives the vertex labels shown in the following table.

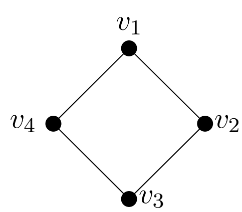
v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}
x_1	x_2	x_4	$[x_2, 4]$	$[x_2, 7]$	x_3	$[x_2, 3]$	$[x_4, 7]$	$[x_3, 5]$	$[x_1, \infty]$	$[x_3, 4]$

List the edge(s) of N which could be added to T_4 in the next iteration and, for each such edge, give a table showing the new vertex labels.

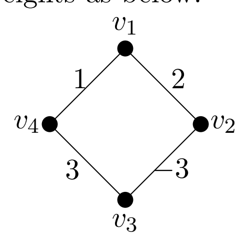
Next edge to be added is v_2v_7. New vertex labels are:

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}
x_1	x_2	x_4	$[x_2, 4]$	$[x_5, 4]$	x_3	x_5	$[x_5, 6]$	$[x_3, 5]$	$[x_1, \infty]$	$[x_3, 4]$

Q2 *Show that Dijkstra's algorithm for constructing a shortest path spanning tree rooted at a vertex v_1 in a network may not work if the network is allowed to have edges with negative weights by assigning suitable weights to the edges in the following network.*

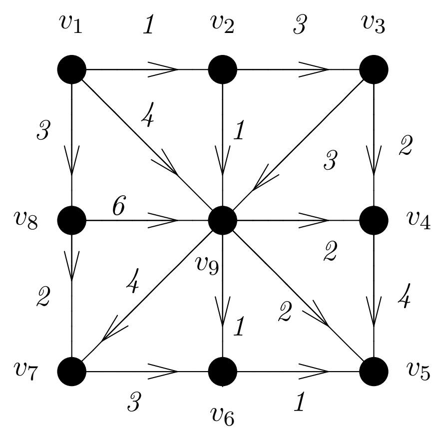


Assign edge weights as below.



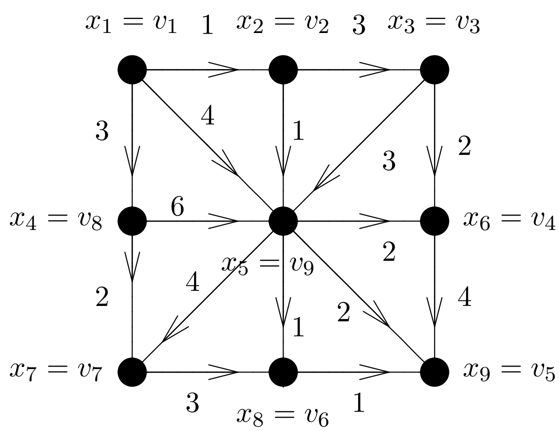
Then Dijkstra's algorithm produces the spanning tree T with $E(T) = \{v_1v_4, v_1v_2, v_2v_3\}$. The path in T from v_1 to v_2 is $P = v_1v_2$ which has length 2. However the shortest path in N from v_1 to v_2 is $P = v_1v_4v_3v_2$, which has length 1.

Q3 *Consider the following acyclic directed network N.*

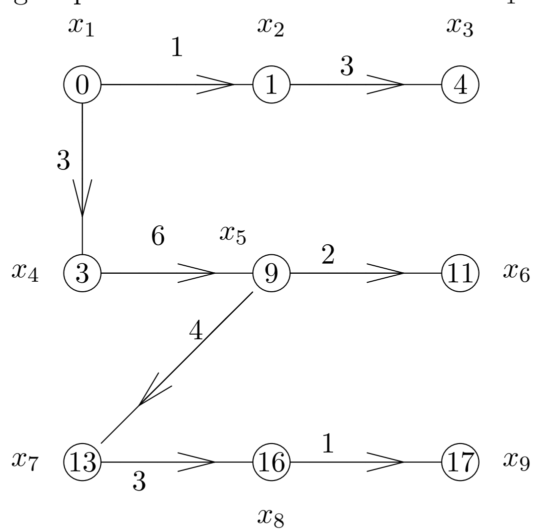


(a) Construct an acyclic labeling of the vertices of N.
(b) Use your acyclic labeling to find a spanning out-arborescence of N rooted at v which contains longest directed paths from v_1 to every vertex of N.
(c) Use your acyclic labeling to find a spanning out-arborescence of N rooted at v which contains shortest directed paths from v_1 to every vertex of N.

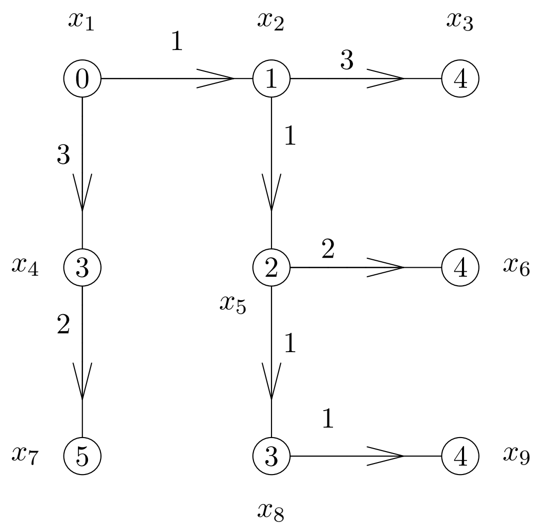
(a) Acyclic labelling is shown below.



(b) Spanning longest path out-arborescence rooted at v_1 is



(c) Spanning shortest path out-arborescence rooted at v_1 is



Q4 *Let N be an acyclic directed network and v be a vertex of N such that N contains a directed path from v to every vertex of N. Let T_i be an out-arborescence rooted at v produced in the i'th iteration of Morávek's algorithm for finding longest paths applied to N.*
(a) Prove that the unique path in T_i from v to each vertex x of T_i is a longest path in N from v to x.
(b) Does your proof assume that N has positive edge weights?

(a) **Proof** Let $long_N(x_1, x_i)$ denote the length of a longest directed path in N from x_1 to x_i, for all $1 \leq i \leq n$. We need to show that $dist_{T_i}(x_1, x) = long_N(x_1, x)$ for all $x \in V(T_i)$. We use induction on i.
Base Case $i = 1$. Since T_1 is an out arborescence with one vertex $x_1 = v$ and no edges, we have $dist_{T_1}(v, x_1) = 0 = long_N(v, x_1)$.
Induction Hypothesis Suppose $i \geq 2$ and that $dist_{T_{i-1}}(x_1, x) = long_N(x_1, x)$ for each vertex x of T_{i-1}.
Inductive Step We have $T_i = T_{i-1} + x_i + x_j x_i$ where $x_j \in V(T_{i-1})$, $x_j x_i \in A(N)$, and x_j is chosen such that $dist_{T_{i-1}}(v, x_j) + w(x_j x_i)$ is as large as possible. Since $dist_{T_i}(x_1, x) = dist_{T_{i-1}}(x_1, x) = long_N(x_1, x)$ for all $x \in V(T_{i-1})$, it suffices to show that $dist_{T_i}(x_1, x_i) = long_N(x_1, x_i)$. We have

$$long_N(x_1, x_i) \geq dist_{T_{i-1}}(x_1 x_j) + w(x_j x_i), \tag{1}$$

since there is a directed path in N from x_1 to x_i of length $dist_{T_{i-1}}(x_1, x_j) + w(x_j x_i)$. On the other hand, if $P = x_1 u_1 u_2 \ldots u_r x_i$ is a longest directed path in N from x_1 to x_i, then, since $u_r x_i \in A(N)$ and $x_1, x_2, \ldots, x_n$ is an acyclic labelling of N, we must have $u_r = x_k$ for some $1 \leq k \leq i-1$. Let $P' = x_1 u_1 u_2 \ldots u_r$. Then

$$\begin{aligned} long_N(x_1, x_i) = w(P) = w(P') + w(x_k x_i) &\leq long_N(x_1, x_k) + w(x_k x_i) \\ &= dist_{T_{i-1}}(x_1, x_k) + w(x_k x_i), \end{aligned} \tag{2}$$

by the induction hypothesis, since $x_k \in V(T_{i-1})$. In the i'th step in Morávek's algorithm we chose the vertex x_j so that $dist_{T_{i-1}}(x_1, x_j) + w(x_j x_i)$ was as large as possible. Thus

$$dist_{T_{i-1}}(x_1, x_j) + w(x_j x_i) \geq dist_{T_{i-1}}(x_1, x_k) + w(x_k x_i). \tag{3}$$

Combining (1), (2) and (3), we obtain

$$\begin{aligned} long_N(x_1, x_i) \geq dist_{T_{i-1}}(x_1 x_j) + w(x_j x_i) &\geq dist_{T_{i-1}}(x_1, x_k) + w(x_k x_i) \\ &\geq long_N(x_1, x_i). \end{aligned}$$

Hence equality must hold throughout, and in particular

$$long_N(x_1, x_i) = dist_{T_{i-1}}(x_1 x_j) + w(x_j x_i) = dist_{T_i}(x_1 x_i).$$

(b) The above proof does not assume that arc weights are positive.