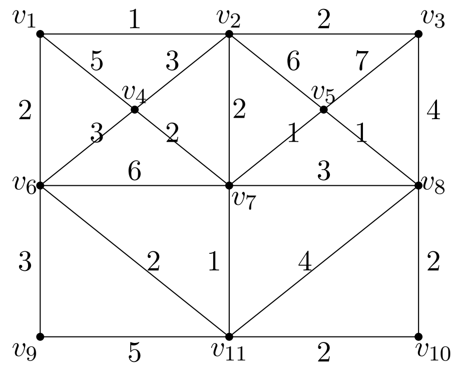
MAS210 Graph Theory Exercises 3 Solutions

Q1 Consider the following network N.

$N =$



(a) *An implementation of Prim'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_3	$[x_4, 3]$	$[x_2, 6]$	x_4	$[x_2, 2]$	$[x_3, 4]$	$[x_4, 3]$	$[x_1, \infty]$	$[x_4, 2]$

List the edge(s) of N *which could be added to* T_5 *in the next iteration and, for each such edge, give a table showing the new vertex labels.*

Possibel edges are v_2v_7 and v_6v_{11}.
If we add v_2v_7 then the new vertex labels will be:

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_3	$[x_5, 2]$	$[x_5, 1]$	x_4	x_5	$[x_5, 3]$	$[x_4, 3]$	$[x_1, \infty]$	$[x_5, 1]$

If we add v_6v_{11} then the new vertex labels will be:

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_3	$[x_4, 3]$	$[x_2, 6]$	x_4	$[x_5, 1]$	$[x_3, 4]$	$[x_4, 3]$	$[x_5, 2]$	x_5

(b) *An implementation of Kruskal's algorithm produces the following forest* F_7 *at the end of the seventh iteration:* $V(F_7) = V(N)$, *and* $E(F_7) = \{v_1v_2, v_5v_7, v_5v_8, v_7v_{11}, v_8v_{10}, v_2v_3, v_6v_{11}\}$. *List the edge(s) of* N *which could* *be added to F_7 in the next iteration.*

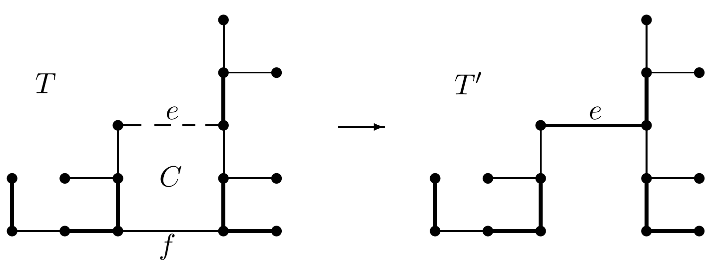
Possibel edges are v_1v_6, v_4v_7 and v_2v_7.

Q2 *Let N be a network and F_i be a forest produced in the i'th iteration of Kruskal's algorithm applied to N. Prove that F_i is contained in a minimum weight spanning tree of N.*

Proof We use induction on i.
Base Case $i = 1$. Since F_1 is a spanning forest of N with no edges, it is contained in all (minimum weight) spanning trees of N.
Induction Hypothesis Suppose $i \geq 2$ and that F_{i-1} is contained in a minimum weight spanning tree of N.
Inductive Step Let T be a minimum weight spanning tree of N containing F_{i-1}. We have $F_i = F_{i-1} + e$ for some edge e of N. If $e \in E(T)$ then F_i is contained in T and we are done. Thus we may suppose that $e \notin E(T)$. Let $H = T + e$. Since $H - e = T$ is connected, e is not a bridge of H. Thus e is contained in some cycle C of H. Let f be an edge of C which does not belong to F_{i-1}. Note that f must exist since otherwise we would have $C - e \subseteq F_{i-1}$ and hence $C \subseteq F_i$, which is impossible since F_i is a forest. Let $T' = H - f$. (So $T' = T + e - f$). Since f is contained in a cycle C of H, f is not a bridge of H. Thus T' is connected. Since

$$|E(T')| = |E(T)| = |V(T)| - 1 = |V(N)| - 1 = |V(T')| - 1,$$

T' is also a spanning tree of N. Furthermore $F_i \subseteq T'$. We complete the proof by showing that T' is a minimum weight spanning tree of N.



F_{i-1} in thick lines $\qquad$ F_i in thick lines

In the i'th step of Kruskal's algorithm, we chose the edge e as an edge of $E(N) - E(F_{i-1})$ such that $F_{i-1} + e$ contains no cycles and, subject to this condition, $w(e)$ is as small as possible. We have $f \in E(N) - E(F_{i-1})$, and, since $F_{i-1} \subseteq T$, $F_{i-1} + f$ contains no cycles. Hence, we must have

$w(f) \geq w(e)$. Thus $w(T') = w(T) - w(f) + w(e) \leq w(T)$. Since T is a minimum weight spanning tree of N, we must have $w(T') = w(T)$ and T' is another minimum weight spanning tree of N.

Deduce that the output of Kruskal's algorithm is indeed a minimum weight spanning tree of N.

Proof The output from Kruskal's algorithm is a spanning forest of N with $|V(N)| - 1$ and hence is a spanning tree T^* of N. It follows from the above that T^* is contained in a minimum weight spanning tree T for N. Since both T^* and T are spanning trees of N we must have $T^* = T$.

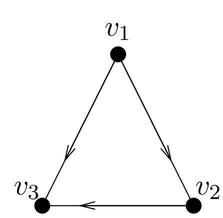
Q3 Let N be a directed network and v be a vertex of N such that every vertex of N can be reached from v_1 by a directed path. An *out-arborescence of N rooted at v* is a directed tree T in N which contains a directed path from v to every vertex of T (so all edges in T are directed 'away from' v). We could try to modify Prim's algorithm to find a minimum weight spanning out-arborescence of N rooted at v as follows.

Initial Step Put $x_1 := v$ and let T_1 be the arborescence with $V(T_1) = \{x_1\}$ and $E(T_1) = \emptyset$.

Iterative Step Suppose we have constructed an arborescence T_i with $V(T_i) = \{x_1, x_2, \ldots, x_i\}$ for some $i \geq 1$.

- If $V(T_i) \neq V(N)$ then choose an arc e of N from a vertex x_j of T_i to a vertex y of $N - T_i$ such that $w(e)$ is as small as possible. Put $x_{i+1} = y$ and $T_{i+1} := T_i + x_{i+1} + e$.
- If $V(T_i) = V(N)$ then STOP. Put $T = T_i$ and output T.

Choose weights for the arcs in the following digraph to get a directed network N for which the above algorithm does not give a minimum weight spanning arborescence rooted at v_1.



Let $w(v_1v_2) = 3, w(v_1v_3) = 2, w(v_2v_3) = 1$. Then the above algorithm constructs the spanning out-arborescence T^* with $A(T^*) = \{v_1v_3, v_1v_2\}$ and

$w(T^*) = 5$. However the minimum weight spanning out-arborescence T of N has $A(T) = \{v_1v_2, v_2v_3\}$ and $w(T^*) = 4$.

Q4 (a) *Let G be a connected graph and H be a connected spanning subgraph of G with as few edges as possible. Prove that H is a spanning tree of G.*

Proof Let e be an edge of H. Then $H - e$ must be disconnected (otherwise it would be a connected spanning subgraph of G with fewer edges than H. Thus every edge of H is a bridge. Thus H contains no cycles and hence H is a tree.

(b) *Let F be a connected graph such that $|E(F)| = |V(F)| - 1$. Prove that F is a tree.*

Proof Let T be a spanning tree of F. (We know that all graphs have a spanning tree by (a)). Then

$$|E(T)| = |V(T)| - 1 = |V(F)| - 1 = |E(F)|$$

Since $E(T) \subseteq E(F)$ we have $E(T) = E(F)$ and hence F is a tree.