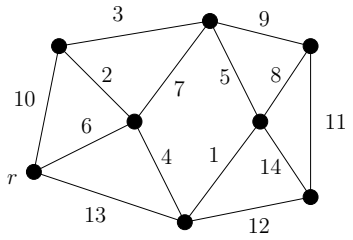
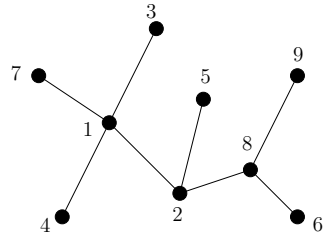
Homework 4 Solutions

Problem 1. In the weighted graph from the figure below, find the sequence of edge weights selected when both Kruskal's algorithm is run, and when Dijkstra's algorithm is run.

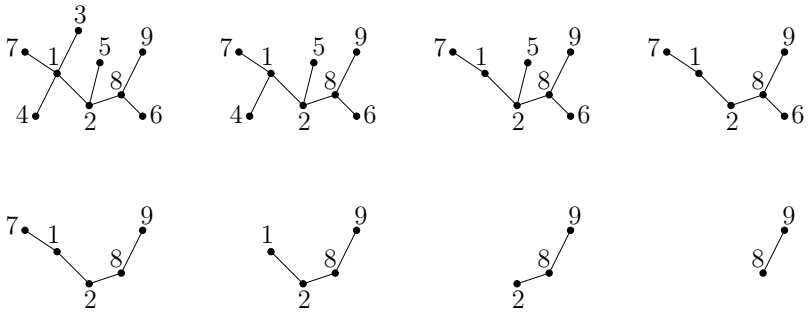


Solution: Kruskal's algorithm chooses the edges in the order given by: $1, 2, 3, 4, 6, 8, 11$. Dijkstra's algorithm for the vertex r gives us the edge sequence: $6, 2, 4, 1, 3(\text{or } 3, 1), 8, 12$.

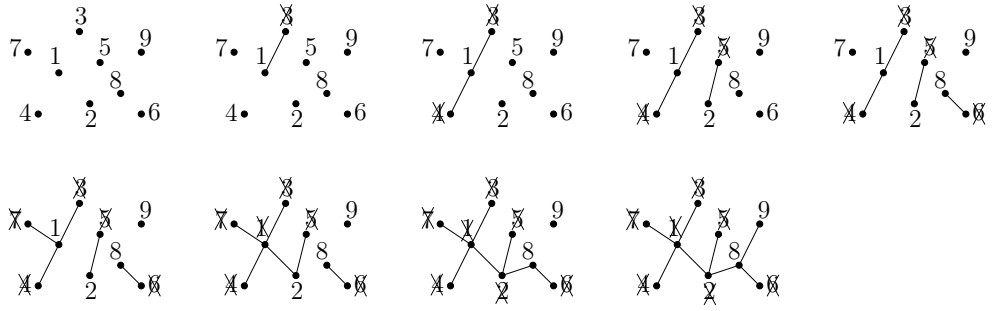
Problem 2. In the tree below with vertex set $\{1, 2, \ldots, 9\}$ apply Prüfer's algorithm to encode this tree, and then apply it to decode this sequence.



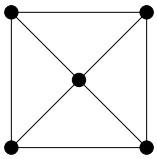
Solution: Prüfer's algorithm encodes this tree as $1, 1, 2, 8, 1, 2, 8$ by the following sequence of trees.



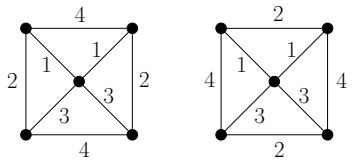
Prüfer's algorithm then decodes this sequence to the original tree as follows:



Problem 3. Assign the weights $1, 1, 2, 2, 3, 3, 4, 4$ to the edges of the graph below in two ways: one way so that the minimum weight spanning tree is unique, and another way so that the minimum weight spanning tree is not unique.



Solution:



In the weighting on the left, the edges of weights 1 and 2 form the unique min cost spanning tree. In the weighting on the right, the two edges of weight 1 together with the "bottom" edge of weight 2 together with either of the edges of weight 3 form a min cost spanning tree.

Problem 4. Let $T = (V, E)$ be a tree with $|V(T)|$ even, and define

$$S = \{e \in E \mid T \setminus e \text{ has two components with an odd number of vertices}\}.$$

Show that every vertex in the graph (V, S) has odd degree.

Solution: Let $v \in V$ and let $T_1, \ldots, T_k$ be the components of the graph $T - v$. For every $1 \leq i \leq k$ let e_i be the edge with one end v and the other in the component T_i. Now $\sum_{i=1}^{k} |V(T_i)| = |V(T)| - 1$ is odd. So, there must be an odd number of components of $T - v$

which have odd size. Since the edge e_i will be included in S if and only if $|V(T_i)|$ is odd, S contains an odd number of edges incident with v. Thus, v has odd degree in (V, S), as desired.

Problem 5. Let G be a connected graph with weight function $w : E(G) \to \mathbb{R}_+$ and assume that w is one-to-one. If $C \subseteq G$ is a cycle and $e \in E(C)$ is the heaviest edge in C, prove that no minimum weight spanning tree contains e. Use this to prove that the following algorithm produces a minimum weight spanning tree: Iteratively delete the highest weight non-cut-edge until the resulting graph is acyclic.

Solution: Suppose (for a contradiction) that T is a spanning tree of minimum weight, and there exists a cycle C so that e is the heaviest edge in C and $e \in E(T)$. Let e have ends u, v and let $P = C - e$ (so P is a path from u to v. Now $T - e$ has two components, say T_1, T_2 and one contains u while the other contains v. Since P is a path from u to v there must exist an edge $f \in E(P)$ with one end in T_1 and the other in T_2. By assumption we have $w(f) < w(e)$, but then $T - e + f$ is a spanning tree with weight less than T, which gives us a contradiction.

For the second part, we repeatedly apply the given algorithm. Since we only delete non-cut-edges, the resulting graph stays connected. Since we stop when the graph becomes acyclic, the output will indeed be a spanning tree. Since the only edges removed were edges which were not contained in any minimum weight spanning tree, the resulting tree must have minimum weight.

Problem 6. Let G be a connected graph on n vertices, and let $\mathcal{T}$ be the set of all spanning trees of G. Construct a new graph G' with vertex set $\mathcal{T}$ where $T_1, T_2 \in \mathcal{T}$ are adjacent (as vertices of G') if $|E(T_1) \cap E(T_2)| = n - 2$. Prove that G' is connected.

Solution: Let $T_1, T_2 \in \mathcal{T}$. We will prove that there is a walk from T_1 to T_2 in the graph G' by induction on $|E(T_1) \setminus E(T_2)|$. As a base case, note that when $|E(T_1) \setminus E(T_2)| = 0$ we have $T_1 = T_2$ so this walk exists trivially. Now for the inductive step we may choose an edge $e \in E(T_1) \setminus E(T_2)$. Let C be the fundamental cycle of e with respect to T_2 and define $P = C - e$. As in the previous problem, since P is a path from one end of e to the other, there must exist an edge $f \in E(P)$ which has one end in each component of $T_1 - e$. Now $T_1' = T_1 - e + f$ is a tree which is adjacent to T_1 in the graph G'. Furthermore, $|E(T_1') \setminus E(T_2)| < |E(T_1) \setminus E(T_2)|$ so by induction, there is a walk from T_1' to T_2 in G'. It

follows from this that there is a walk from T_1 to T_2 in the graph G' as desired.

Problem 7. Let T be a tree and let $T_1, \ldots, T_m \subseteq T$ be trees with the property that $V(T_i) \cap V(T_j) \neq \emptyset$ for every $1 \leq i, j \leq m$. Prove that there is a vertex v contained in $V(T_i)$ for every $1 \leq i \leq m$. Hint: induction on $|V(T)|$.

Solution: We proceed by induction on $|V(T)|$. If $|V(T)| = 1$ then the result holds trivially. For the inductive step we have $|V(T)| \geq 2$ so we may choose a leaf vertex $v \in V(T)$, and we let u denote the unique neighbour of v. We consider two cases. If there is a tree T_j which contains only the vertex v, then we must have $v \in V(T_i)$ for every $1 \leq i \leq m$ so we are done. Otherwise, we define $T' = T - v$ and $T'_i = T_i - v$ for every $1 \leq i \leq m$. By our assumptions, every T'_i is a subtree of T'. Since no tree T_i consisted only of the vertex v, every T_i which contains the vertex v must also contain the vertex u. But then $V(T_i) \cap V(T_j) \neq \{v\}$ for every $1 \leq i, j \leq m$. It follows from this that $V(T'_i) \cap V(T'_j) \neq \emptyset$ for every $1 \leq i, j \leq m$. However, then by induction the tree T' contains a vertex w which is contained in every T_i and it follows that w is contained in every tree T_i.