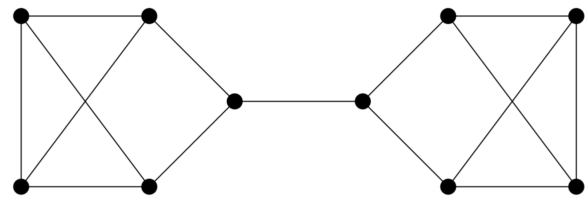
# Math 239 Spring 2014 Assignment 8 Solutions

1. (a) {2 marks} Draw a 3-regular graph that has a bridge.

   **Solution.**



   (b) {4 marks} Prove that any 3-regular bipartite graph cannot have a bridge.

   **Solution.** Let $G$ be a 3-regular bipartite graph with bipartition $(A, B)$. Suppose $e = uv$ is a bridge where $u \in A$ and $v \in B$. Let $C$ be the component of $G - e$ that contains $u$. Now every vertex in $V(C) \cap A$ has degree 3, except for $u$ which has degree 2. And every vertex in $V(C) \cap B$ has degree 3. Since $C$ is a bipartite subgraph, the sum of degrees in $V(C) \cap A$ is the same as the sum of degrees in $V(C) \cap B$. So $3(|V(C) \cap A| - 1) + 2 = 3|V(C) \cap B|$, or $2 = 3(|V(C) \cap B| - |V(C) \cap A| + 1)$. But this is not possible since 2 is not a multiple of 3. So $G$ cannot have a bridge.

2. {3 marks} Let $T$ be a tree on $n$ vertices where every vertex has degree 1 or 4. Prove that $n \equiv 2 \pmod 3$.

   **Solution.** Supper $T$ has $x$ vertices of degree 4. Then there are $n-x$ vertices of degree 1. So the sum of all vertex degrees is $4x + (n - x) = 3x + n$. Since there are $n - 1$ edges in $T$, by Handshaking Lemma, $2(n - 1) = 3x + n$, which means that $n = 3x + 2$. This implies that $n \equiv 2 \pmod 3$.

3. {4 marks} Let $G$ be a connected graph, and let $e \in E(G)$. Prove that $e$ is a bridge if and only if every spanning tree of $G$ contains $e$.

   **Solution.** We will prove the contrapositive in both directions.

   ($\Rightarrow$) Suppose there exists a spanning tree $T$ that does not contain $e$. Then $T$ is a subgraph of $G - e$, so $T$ is a spanning tree of $G - e$. Then $G - e$ is connected, meaning $e$ is not a bridge.

   ($\Leftarrow$) Suppose $e$ is not a bridge. Then $G - e$ is connected, hence it has a spanning tree $T$. So $T$ is a spanning tree of $G$ not containing $e$.

4. {4 marks} Let $T$ be a tree with $k$ edges, and let $G$ be a graph where every vertex has degree at least $k$. Prove that $T$ is a subgraph of $G$. (In particular, this implies that in a graph with minimum degree $k$, you can find a copy of every tree with $k$ edges.)

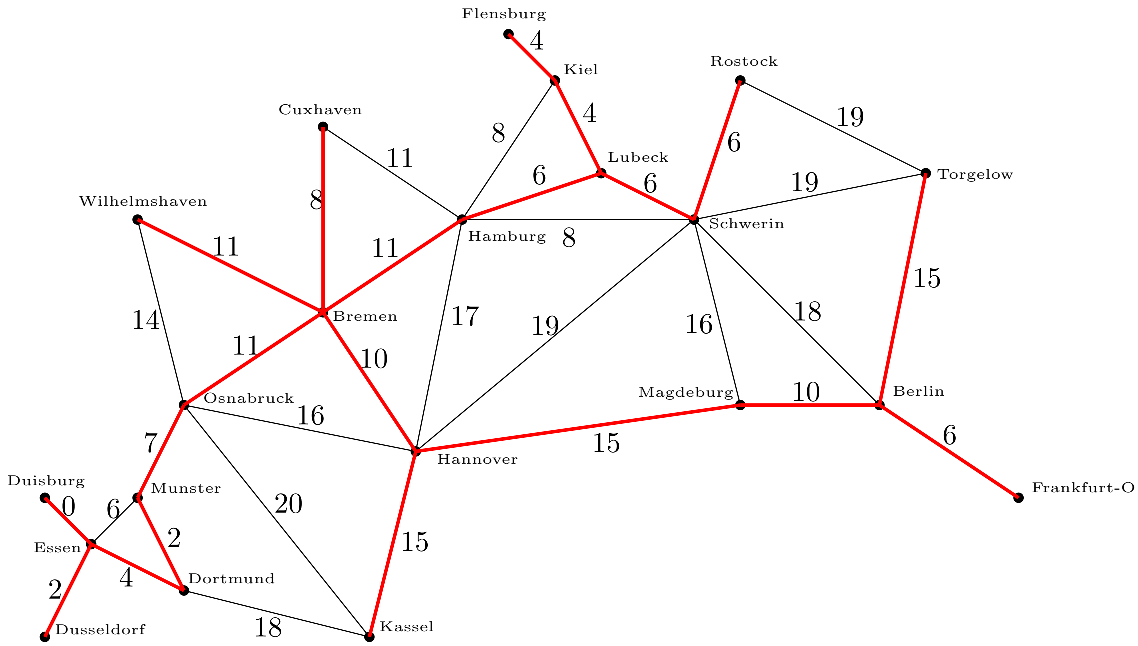
   **Solution.** We prove by induction on $k$. When $k = 1$, $T$ consists of just 2 vertices and 1 edge, while $G$ has a vertex of degree at least 1, so it must contain an edge. Therefore, $T$ is a subgraph of $G$.

   Assume that any graph with minimum degree $k - 1$ contains any tree with $k - 1$ edges.

   Let $G$ be a graph with minimum degree $k$. Let $T$ be any tree with $k$ edges. Let $v$ be a leaf in $T$, and let $u$ be its only neighbour. Then $T - uv$ is a tree with $k - 1$ edges. Since every vertex in $G$ has degree at least $k - 1$, by induction hypothesis, $G$ contains $T - uv$ as a subgraph. Suppose the vertex $u$ in $T - uv$ is mapped to the vertex $x$ in $G$. Since $x$ has degree at least $k$ and $T - uv$ has $k - 1$ edges, at least one neighbour $y$ of $x$ is not used in $T - uv$. Therefore, we can map $v$ to $y$, edge $uv$ to $xy$, and this completes the tree $T$ in $G$.

5. {4 marks} Produce a minimum spanning tree of the following graph using Prim's algorithm. Use Flensburg as your starting vertex. If you have a choice of new cities to add, use the city whose name is alphabetically first. You do not need to show your work. (Source: A portion of the map of Germany from the board game Power Grid.)

   **Solution.**




6. {4 marks} Let $G$ be a connected graph with given edge weights. Prove that the heaviest edge in every minimum spanning tree of $G$ has the same weight.

**Solution.** Let $T$ and $T'$ be two MSTs whose heaviest edges have different weights. Wlog, assume that the heaviest edge in $T$ is heavier than the heaviest edge in $T'$. Let $e \in E(T)$ be an edge in $T$ with the heaviest weight. Then $T - e$ has two components. Since $T'$ is a spanning tree, there exists an edge $e' \in E(T')$ such that $e'$ is in the cut induced by the vertices of one component. Therefore, $T - e + e'$ is a spanning tree. However, since the weight of $e$ is heavier than the heaviest weight in $T'$, $w(e) > w(e')$. So $w(T - e + e') = w(T) - w(e) + w(e') < w(T)$, which is a contradiction since $T$ is a MST.

7. {3 marks} Prim's algorithm for the minimum spanning tree problem is a greedy algorithm, meaning we take the best possible edge to add at each step, and finish with an optimal solution. But greedy algorithms do not always work. Consider the problem of finding a minimum spanning path in a weighted complete graph $K_n$. Start with an edge of minimum weight. At each step, say we have a path $P$ with endpoints $u, v$. Consider all edges joining $u$ or $v$ with a vertex outside of $P$. Pick one that has the smallest weight and extend our path by one edge. We repeat until we get a spanning path.

Find an example for $K_4$ where the algorithm above does not produce a minimum spanning path. Clearly indicate the path produced by the algorithm, and an optimal spanning path.

**Solution.** Consider the following weighted $K_4$. The red path is one produced by the algorithm, which has weight 162. The blue path is a minimum weight spanning path, which has weight 8.