MATH 239 Spring 2014: Assignment 8

Due: 3:00 PM, Monday July 14, 2014 in the dropboxes outside MC 4066

Last Name: First Name:

I.D. Number: Section:

Mark (For the marker only): /28

---

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

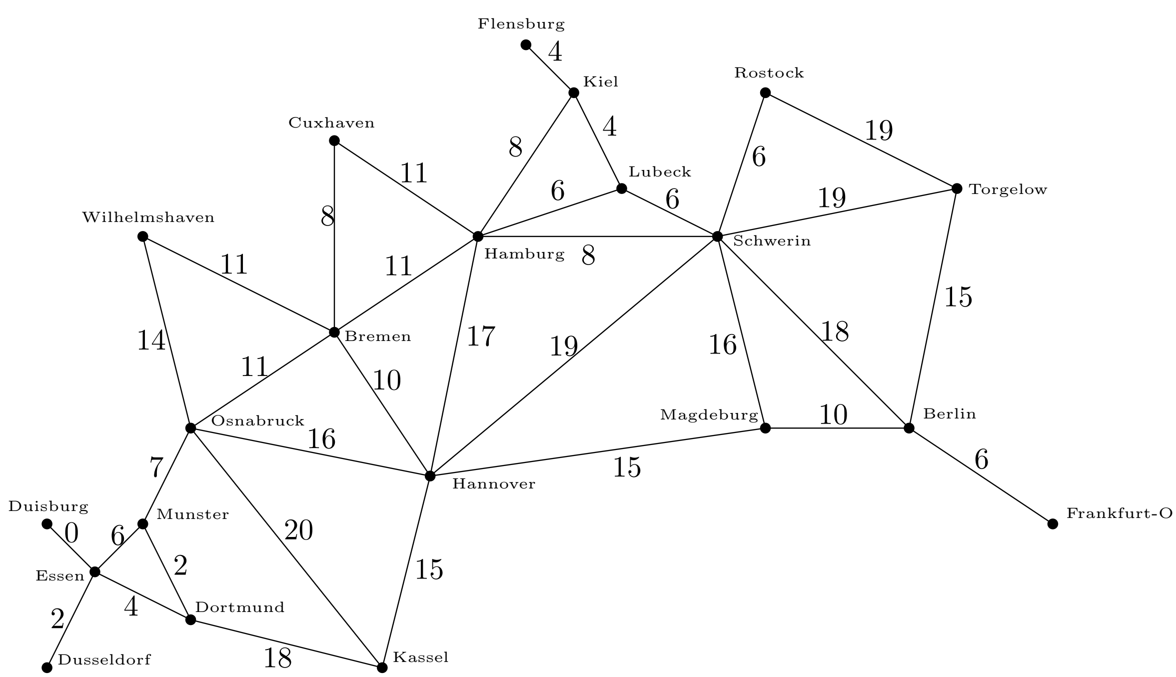
   (b) {4 marks} Prove that any 3-regular bipartite graph cannot have a bridge.

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

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.

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.)

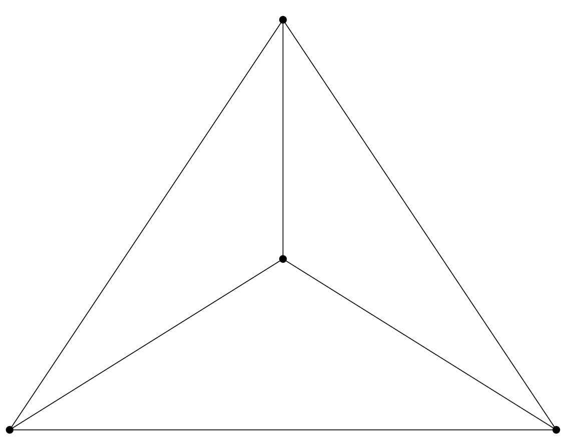
5. {4 marks} Produce a minimum spanning tree of the following graph using Prim's algorithm. Use Flensburg as your starting vertex. If you have a choice of new cities to add, use the city whose name is alphabetically first. You do not need to show your work. (Source: A portion of the map of Germany from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



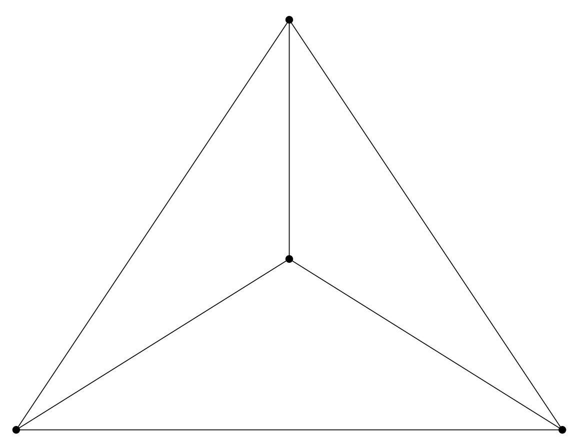
6. {4 marks} Let G be a connected graph with given edge weights. Prove that the heaviest edge in every minimum spanning tree of G has the same weight.

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.



Spanning path from the algorithm



Shortest spanning path