1. Which vertices in a flow problem have their incoming flow not equal to outgoing flow? Explain. What is a capacity, what is flow on an edge, what is residual capacity on an edge? (answer for the last: capacity that remains after a flow is allocated on an edge, computed as \( c(u, v) - f(u, v) \))

2. What does NP stand for?

3. If I were to use a linked list rather than a heap as my main data structure in computing the minimum spanning tree, what would my running time be? Analyze. What if I used an array? Think of other types of data structures and analyze how this algorithm would behave if they were used.

4. Half-cycle. Given a graph \( G \) with \( n \) nodes, does it contain a simple cycle of length at least \( n/2 \)? Prove that this question is NP-complete.

5. Look up the traveling salesman problem and prove that it is NP-complete. What did you reduce from?


7. I give you a graph where all edges have the same weight, \( w \). Give me an algorithm to compute the cost of the minimum spanning tree in constant time. How about computing the actual tree – how fast can you do that?

8. Let \( G \) be a graph, and edge \( e \) be an edge in \( G \). \( e \) has the smallest weight in the entire graph; all other edges have higher weight. Prove that any MST of \( G \) must contain \( e \). This is a hard question, think about perhaps proving by contradiction.