COSC 311: ALGORITHMS
MINI 11
Solutions

1. Suppose I've given a polynomial time reduction from problem V to problem W (that is, I've shown that $V \leq_P W$). Fill in the blanks:

If $\underline{V}$ cannot be solved in polynomial time, then neither can $\underline{W}$.

Explain why the above statement is true.

Solution: If V cannot be solved in polynomial time, then every single possible algorithm to solve V must take exponential time. One possible algorithm to solve V is:

1. Reduce V to W
2. Solve W

This algorithm must take exponential time. However since $V \leq_P W$, we know that step 1 takes polynomial time, so step 2 must take exponential time. Hence there cannot exist a polynomial-time algorithm to solve W.

(Note that on the other hand, if W cannot be solved in polynomial time we do not learn anything about how long it might take to solve V. Reducing V to W and then solving W is just one possible algorithm to solve V; there could be some other algorithm that has nothing to do with W and that takes polynomial time.)

2. Suppose I have a problem Z that I want to prove is NP-complete. What are the two things I'd have to do to show this?

Solution: We'd have to show:

1. Z is in NP, meaning that there exists a polynomial-time algorithm to verify whether an instance of Z is a yes-instance. To do this, we can describe an algorithm that does the verification (and analyze its runtime to show that it takes polynomial time).
2. Z is NP-hard, meaning that for all problems Y in NP, $Y \leq_P Z$. To do this, we can take a problem that we already know is NP-hard and reduce that problem to Z. For example, we know that Circuit-SAT is NP-hard. If we show that Circuit-SAT $\leq_P Z$, then we have that for all problems Y in NP, $Y \leq_P$ Circuit-SAT $\leq_P Z$, so by transitivity $Y \leq_P Z$.