Group Work on Satisfiability

Learning Objectives:

1. To strengthen our understanding of propositional logic
2. To practice algorithm development and analysis

The problem: Satisfiability

- **Input:** A compound logical expression.
- **Output:**
  - "Yes" if there is a way to assign truth values to the variables to make the expression true.
  - "No" if not.

Part I: Understanding the problem. Figure out the truth tables for each of the following compound logical expressions and then decide what the output would be for each.

1. $((\neg p) \to (\neg q)) \wedge (p \to q)$
2. $((\neg p) \to (\neg q)) \wedge \neg(q \to p)$

Part II: Developing an algorithm.

1. Develop an algorithm to solve this problem based on truth tables. You should give enough detail so that if you told the algorithm to your 8 year old little sister, and you were to then quiz her, she would always get the right answer by following your directions to the letter. How should she label the columns, exactly? How should she set up the rows? In what order should she fill in the table? At what point can she stop and output the answer? You may assume that she knows how to figure out if a logical expression involving T and F is true; for example, she knows that $F \to F$ is T.

Part III: Analyzing the speed of the algorithm.

1. How many rows did your truth table have for the examples above?
2. How many rows will your truth table have in general? What does this depend on?
3. How many columns did your truth table have?
4. How many columns will your truth table have in general? What does this depend on?
5. How many entries did your truth table have for the examples above?
6. If the expression has m variables and n logical operators, how many entries will the truth table have?
7. If the computer can fill in one entry in one one-billionth of a second, how long would it take to fill in the truth table for the examples above?
8. And in general, if there are m variables and n logical operators? Your answer here will be a formula $f(m, n)$ in terms of m and n.
9. Find a formula $f(n)$ that expresses the time it takes for the algorithm to run if there are n variables and n logical operators.
10. Suppose that there are 100 variables and 100 logical operators. How long will it take? Express your answer using units (seconds, minutes, etc) that make sense given your answer.
11. **Try this if you're brave:** Can you figure out a faster algorithm?