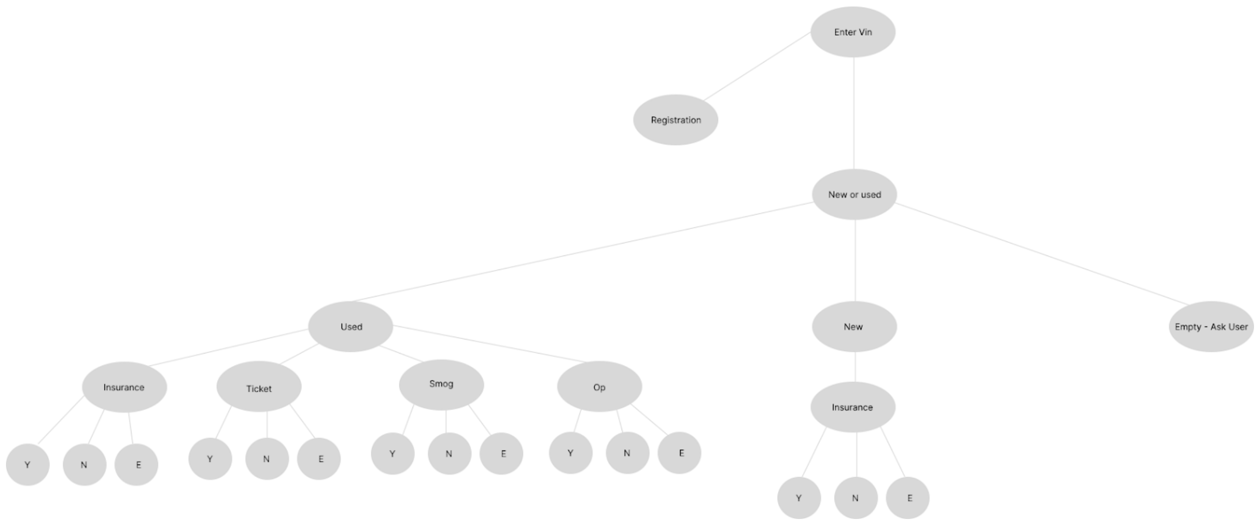
Project 2: DMV

Group Members: Terry Weatherman, Brynn Crowley, Oanh Dingle, Justin Mariano, Emran Zahir, Seyed Mojitaba Karanizadeh, Daniela Pérez

Fundamentals of Algorithmic Problem Solving:

1. **Ascertaining the capabilities of the Computational device**
    a. Our assignment utilizes sequential algorithms.

2. **Understand the Problem**
    a. Our problem implements an object to obtain DMV car registration from users. A user is prompted to input their car registration information. They begin by inputting a VIN number. This will allow us to determine if their information is already documented in our database. The program will then traverse through the tree. The tree has 3 possible outcomes, True, False, or Empty, in which we ask the user to input the information. After the user inputs all of their information a receipt is reported to the user indicating the amount they need to pay.



3. **Choosing between exact and approximate problem-solving**
    a. We wanted to keep our decision tree relatively simple, so we stuck to asking the user 4 questions. We ask for Insurance, Tickets, Smog, and if it Non-Op. Since we also ask the user whether it is a new or used vehicle, new vehicles are only asked for insurance.

4. **Algorithm Design Technique**

    a. In order to get a skeleton for our project, we created a very detailed pseudocode. This also allowed us to have an idea of which methods we wanted to implement into our project.

5. **Designing an Algorithm and Data Structure**
    a. As per the foundation of this project, we use a tree structure. However, we also use an abundance of conditionals and arrays. For example, we use arrays in our carTypes class containing different car models. Additionally, we use if statements that traverse through the arrays to determine the price for registration.

6. **Methods of Specifying Algorithm**
    a. Our program consists of 5 classes. We have a BooleanNode class that creates the tree's root and a BooleanBinaryTree class that creates a binary tree. The tree root and tree are both done recursively. Further, we have the carTypes class that utilizes arrays to store data on different car models. The VehicleClass is where most of the conditionals lie to get information on the user. Finally, there is the main class that is used to run the program and output the result to the user.

7. **Proving an Algorithm is Correct**
    a. For our POC (Proof of concept), we decided to test our ternary boolean system. In order to do this, we did unit testing. Instead of asking the user the questions, we wanted to confirm our tree worked by using only set values and running those through the tree.

8. **Analyzing an Algorithm**
    a. Since we cannot see our program iterating through the tree, we must rely on the output to analyze our algorithms. Starting from the Proof of Concept, we have relied heavily on the output to test our code.

9. **Coding an Algorithm**
    a. https://github.com/Lildel81/Project-2-CSC130