## CS 101 -- Section 1 -- Problem Solving

In class this week we talked about the very basics of what an **algorithm** is. Recall that an **algorithm** is just a series of very detailed steps (perhaps with repetitions) for solving a problem. Computer scientists use algorithms to solve computational problems.

Soon, we will start to learn how to program in a language called **Processing**. When we do write programs, we have to "think algorithmically" because programs have to be told exactly what to do, in a step-by-step manner. Remember: When we are coming up with an algorithms we have our "initial state" (the input) and out "output state" (the result). Below are several problems that people need to solve in real life. Come up with an **algorithm** and corresponding **decision tree** to solve each of these. Remember that **details are important**. Don't just write 1 or two sentences explaining the problem.

(1) Beginning state: Driving down a road
Output state: Parallel parked on the side of the road

(2) Beginning state: In a stocked kitchen, wanting a hamburger Output state: Having a fully cooked hamburger

(3)	Beginning a game of Tic-Tac-Toe with an opponent, you have the first turn Winning the game	
(4)	Beginning a game of Battleship with an opponent, you have the first turn Winning the game	
(4)		
(4)		
(4)		
(4)		

Scratch paper for diagrams and algorithms: