1. **Introduction To Time Complexity**

* A standard that is used as benchmark to evaluate algorithms or compare algorithms
* It uses the big O notation

1. **Math Refresher**

* CS is applied math
* ***The idea of Log Functions (Using it is important in CS, because it does not impact the run time significantly)***
* It is short for logarithmic function, and it is the inverse of an exponential function. *Also, for regular numbers the inverse is 1 dividing by them -> 5’s inverse is 1/5. But for the logarithmic function it is inverse is not 1 divide by it.*
* An exponential function is known as e = 2.7. Therefore e^1 = 2.7, e^2 = 2.7^2 7.29. Thus, e beings by having slight increases between each iteration, then the more iterations the larger the run time will be. In e the resultant will end up being a vertically continuously increasing line. However, for log we beings by having significant changes but the more iterations we have the smaller the changes of the resultant values become and the smaller our run time will be. In log the resultant will end up being a horizontal with extremely slight vertical increasing line
* The log formula -> Log x(y) = b
* Log based x of y is equal to b
* The inverse exponential formula -> x b = y
* X to the power of b is equal to y
* Exp:-> Log 10(5) = 0.69897 & 10 0.6989 = 5
* The log function helps us fund the number b easier
* Exp:-> Facebook friends Exp. Showing an example if the CS algorithm was based on log
* You have 64 friends on FB and we would like to know to the run time
* We have a specific amount of data, and we would like to know the time we need to analyze it. We will use base 2 because the computer always takes base 2
* Log 2(64) = 6
* **Note**: in CS we use log base 2. Which is the value x in the above examples. We use log base 2, because computers have only two states of 0 and 1
* ***The idea of Factorial Functions***
* A factorial is a # followed by!
* The factorial of a number is all the numbers below it multiplied by each other, including the number itself
* 6! Is equal to 6\*5\*4\*3\*2\*1 = 720
* **Note:** the growth rate between each iteration is insane, while using the factorial. You want to avoid factorials and in CS It is represented by n!
* ***The idea of Algebraic Expressions***
* Using n to represent place holders. This is just showing how formulars should look like.
* This an extremely basic lecture, showing how values are represented by variables