## 11. Prove and demonstrate that binary search takes O(log(n)).

Binary Search calls itself/iterates through the while loop a number of times based on what power of 2 is it less than or equal to.

If its 2^1  = 2 elements it's found in one loop
If its 2^2 = 4 elements it'll take 2 loops to find the individual elements when breaking up by 2
If its 2^3 = 8 elements it'll take 3 loops splitting it into two to find each individual
And so on

We're looking for what power of 2, 2^x can be used to divide N to be less than or equal to 1
Aka
1 = n / (2^x)
Multiply both sides
2^x = n
The definition of logarithms would let you do
Log(Base2)(n) = x

So the amount of times the function loops is based on Log(Base2)(n)

Which falls under the O(log(n)) value