Q1: Describe an analogy for relating an algorithm that has efficiency O(1) and another algorithm that has $O(2^n)$.

Imagine Bunnies! If we had a pen of bunnies and every female bunny took birth control. As long as none of them died, we would have a constant number of rabbits.

If we didn't give the female bunny rabbits birth control and penned them up, their population would grow exponentially!

Q2: In plain English, what is the best-case scenario for binary search?

Binary search is searching a sorted array by continually dividing the array in half. Therefore the best-case scenario is if the element we are looking for is the exactly half way in the sorted array!

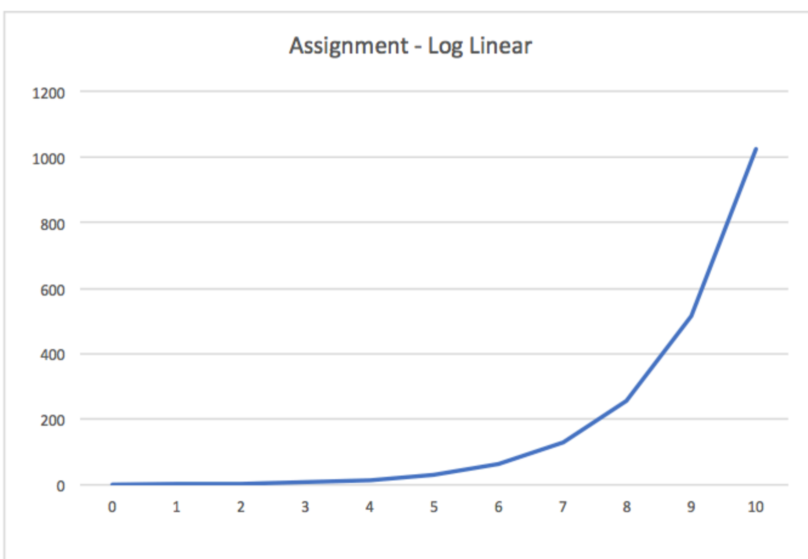
Q3: In plain English, what is the worst-case scenario for binary search?

The worst case is actually if the item wasn't even contained in the array. It would go through the entire array halving the interval until eventually nothing was found.

Q4: In plain English, what is the bounded-case scenario for binary search?

The bounded-case would be that the it does exist and the value we are searching for is somewhere not in the first, middle or last position.

Q5: Create a graph using the data below. Here's a **CSV** with the values you'll need.



Q6: What is the limit of the function above as n approaches infinity?

The function is essentially $f(n) = 2^n$
That means as n approaches infinity the function approaches infinity as well.

Q7: What is the Big-O of an algorithm that has the data points above?

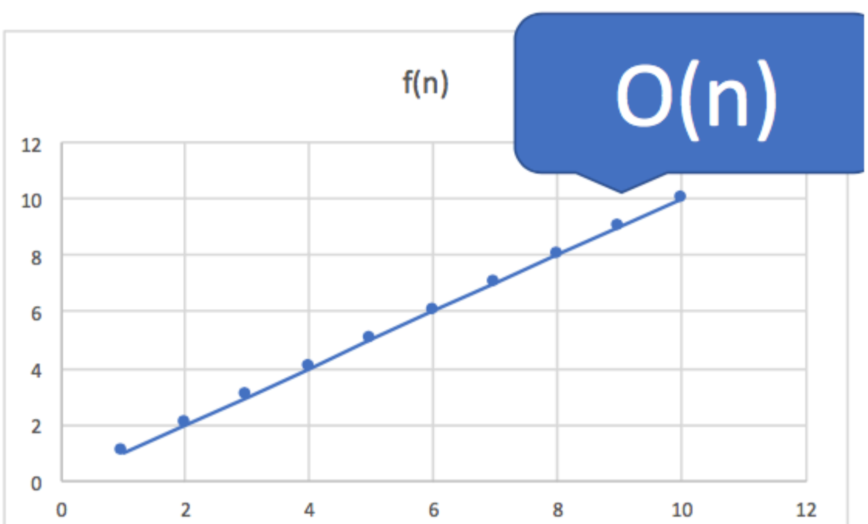
$O(2^n)$

Q8: See worst_case_linear.rb

Q9:

Linear Search

X	**f(n)**
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10



Q10: What is the Big-O of binary search?
Big O of binary search (worst case scenario) is $O(\text{Log } n)$ where n is the number of elements in the array. The worst case is where no element is found.

Q11: What is the Big-Ω of binary search?
Big-Ω of binary search (best case scenario) is $\Omega(1)$ where the item we are looking for is right in the middle.

Q12: What is the Big-Θ of binary search?
Big-Θ of binary search which is the average case, is $\Theta(\text{Log } n)$.