**Whiteboarding #3 Questions**

1. Be sure to follow the process outlined in the Whiteboarding cheat sheet!
2. Everyone should pair up to take turns whiteboarding the following problems and **identifying the efficiency of each algorithm in terms of Big O Notation**.

1. Write a function that reverses a string. Don’t use the “reverse” method!  
  
2. Write a function that accepts a string and returns the number of times that the letter “a” occurs in it.  
  
3. Write a method that accepts two arguments. The first argument is an array of numbers that are in ascending order. The second argument is a number to search for within in the array. The method should do a linear search and return the index at which this value is found, or it should return nil if the value is not found. See if there’s a way in which you can optimize this method to keep its number of steps to a minimum! Note: Do not use the “index” method!  
  
4. Write a method that accepts two arguments. The first argument is an array of numbers that are in ascending order. The second argument is a new number that is not yet in the array. The method should return a new array with the new number inserted in the proper place. Do not use the “sort” method!  
  
5. Write a method that accepts two arguments. The first argument is an array of numbers that are in ascending order. The second argument is a number that is contained within the array. The method should return the string “lower” if the value is found in the lower half of the array, and it should return “higher” if the value is found within the greater half of the array. Try to optimize this algorithm so that it takes a minimum number of steps!  
  
6. Work on this one together: Write a method that does binary search. Specifically, it should accept two arguments. The first argument is an array of numbers that are in ascending order. The second argument is a number that may or may not be contained within the array. The method should perform binary search to find the index at which this second value is found within the array. If the value cannot be found within the array, the method should return nil.

7. Write a method that accepts two arrays, and returns a new array that does a “riffle shuffle” between them. That is, if the first array is [1, 3, 5, 7, 9], and the second array is [10, 8, 6, 4, 2], the returned array should be [1, 10, 3, 8, 5, 6, 7, 4, 9, 2].

BONUS: Read about and implement the following sorting algorithms that sort an array of numbers. Be sure to identify the Big O efficiency of each algorithm!

\* Bubble sort

\* Insertion sort

\* Selection sort

\* Merge sort

\* Quick sort

\* Heap sort