Isaiah Jenkins

9/13/14

Big O Analysis of Practice Assignment 3

Iterative = O(n)

```
public int find_min_iterative () {

        int min = 0; \\ O(1)

        //runs through the array to find the min index
        for(int i = 0; i <= arr.length - 1; i++) {  \\ O(1)

            if(arr[i] < arr[min]){  \\ O(n)

                min = i;  \\ O(n)

             }

        }

        return min;  \\ O(1)

            }
```

Calculations:

Because the updated index, i, is wrapped in one loop, this function is O(n)

Recursive = O(n)

```
public int find_min_recursive () {

                return findMinRecursive(arr, 0, arr.length - 1);
        }

public int findMinRecursive(double[] theArray, int theI, int arrSize){


     if(theI == arrSize){       // O(1)

       return theI;         // O(1)

     }

     int theMin = findMinRecursive(theArray, theI + 1, arrSize);  // O(n)


     if(theArray[theMin] < theArray[theI]){   // O(1)

          return theMin;   // O(1)

     }


  }
```

Calculations:

Because the index, theI, is the dominant unit in recursion, this function is O(n)
O(theI + 1) = O(theI) = O(n)