find_min_iterative() Big-O (Informal)

```
public int find_min_iterative () {
                int min_index = 0; O(1)
                for(int i = 1; i < arr.length; i++){ O(1)
                        if(arr[i] < arr[min_index]){ O(n)
                                min_index = i; O(n)
                        }
                }return min_index;
        }
```

Even at its best, we see this algorithm is **O(n)** where n is the number of elements in the array

find_min_recursive() Big-O

The helper/overloaded method that calls this function would be O(1). Note the helper method starts the index and minIndex both at 0.

```
public int find_min_recursive(double[] arr, int index, int minIndex, int arr_length){
                if(index == arr_length){
                        //Reached end of array, return the min index
                        return minIndex;
                }else{
                        if(arr[index] < arr[minIndex]){
                                minIndex = index;
                        }
                }return find_min_recursive(arr,index+1,minIndex,arr.length);
        }
```

The runtime of this function can be analyzed by looking at the base case. Since the function is terminated only after the index has reached the end of the array (index == array length), the function would be no better than **O(n).**