Problem 2

```
Bubble sort(int size, int[] arr)
  //choose first element
    Int i= 0;
  //select every element in the array
    For(0<=index<size)
    //compare left element with right, and swap if left element is greater than the right element.
    //compare until the element has been compared to all elements to its right
      For(1<=j<size)
         If(arr[i] >arr[i+j]) swap
```

Nested for loops so O(n^2)

```
Merge sort(int size, int[] arr)
//split array in half into two sub arrays.
    Mid = size/2;
    LeftArr = {arr[0],…arr[mid]}
    RightArr = {arr[mid],…arr[size-1]}
// repeat the process with the sub arrays until they have a size of one.
//use base case as first line of code in this function (this goes before the splitting)
    If size == 1, return
//Choose the left most element in Leftarray and compare it with the leftmost of the RightArray
    For(0<= index < LeftArr.size)
        For(0<= index2 < RightArr.size)
           If(leftArr[index]>rightArr[index2)
             Swap
```

This has a nested loop but also has recursive calls, which handles sub problems.

O(nlogn)