Problem 2, Part B:

```
public static void mergeSort(int[] arr, int x, int y) {

    //Sorting one item
    if (x == y) {
        return;
    }
    //Sorting two items
    if (y - x == 1) {
        if (arr[x] > arr[y]) {
            swappingVals(arr, x, y);
        }
        return;
    }
    //Recursive procedure
    int mid = (x + y) / 2;

    mergeSort(arr, x, mid);
    mergeSort(arr,  x mid + 1, y);
    mergeVals(arr, x, y, mid);
}
```

The merge itself takes $\log(n)$ to do. Since the merge calls the loop each time the complexity would, $O(nlog\ (n))$