Q1.2.

We can still use the similar idea to sort this array using nlogn time. I'll still have a sample code to show how I sort this array using nlogn time.

```
class MergeSort {
    void merge(int arr[], int l, int m, int r) {
        int n1 = m - l + 1;
        int n2 = r - m;
        int l[] = new int[n1];
        int r[] = new int[n2];
        for (int i = 0; i < n1; ++i)
            l[i] = arr[l + i];
        for (int j = 0; j < n2; ++j)
            r[j] = arr[m + 1 + j];
        int i = 0, j = 0;
        int k = l;
        while (i < n1 && j < n2) {
            if (l[i] <= r[j]) {
                arr[k] = l[i];
                i++;
            }
            else {
                arr[k] = R[j];
                j++;
            }
            k++;
        }
        while (i < n1) {
            arr[k] = l[i];
            i++;
            k++;
        }
        while (j < n2) {
            arr[k] = r[j];
            j++;
            k++;
        }
    }
```

```
35 ▾    void sort(int arr[], int l, int r) {
36 ▾        if (l < r) {
37              int m =l+ (r-l)/2;
38              sort(arr, l, m);
39              sort(arr, m + 1, r);
40              merge(arr, l, m, r); // merge the sorted
41          }
42      }
```