```
if(first<last){
   int mid = first + (last-first)/2;

   merger(a,first,mid, arraySize);

   merger(a,mid+1,last, arraySize);


   merge(a,first,mid,last,arraySize);
}
```

Since I am going through the entire array and am dividing it recursively by 2, I have a complexity of $N\log_2 N$. The loop through the entire array results in the linear part of this, while the divisions into groups of 2 results in the logarithmic part. Therefore, our complexity is linearithmic. Since I do not implement a check to see if the array is fully sorted at the start, the best and worst case are both linearithmic.

```
if (a[first1] <= a[first2]) {

     first1++;
   } else {
     temp = a[first1];
     a[first1] = a[first2];
     a[first2] = temp;

     first2++;
   }

}
```

Since I only swap for the greater than case, the code is stable.