

Merge Sort Through Example

AP Computer Science A

Mr Fahrenbacher

6	2	4	5	1
---	---	---	---	---

38	27	43	3	9	82	10
----	----	----	---	---	----	----

10	5	7	6	1	4	8	3	2	9
----	---	---	---	---	---	---	---	---	---

```

public class MergeSort
{
    private static int[] storage;

    public static void sort(int[] array) {
        storage = new int[array.length];
        sort(array, 0, array.length-1);
    }

    public static void sort(int[] array, int lo, int hi) {
        if(lo + 1 == hi) { //sub array with 2 items, might need to swap them!
            if(array[lo] > array[hi])
                swap(array, lo, hi);
        }
        else if (hi > lo) { //sub array with 3 or more items - divide and conquer!
            int firstLo = lo;
            int firstHi = (lo + hi)/2;
            int secondLo = firstHi + 1;
            int secondHi = hi;

            //split into two sub arrays, sort recursively, then merge together
            sort(array, firstLo, firstHi);
            sort(array, secondLo, secondHi);
            merge(array, firstLo, firstHi, secondLo, secondHi);
        }
    }

    //basic swap code
    public static void swap(int[] array, int i, int j) {
        int temp = array[i];
        array[i] = array[j];
        array[j] = temp;
    }
}

```

```
//Merge Sort class continued!
```

```
public static void merge(int[] array, int lo1, int hi1, int lo2, int hi2) {  
    int currentIndex = lo1;  
    int i = lo1;  
    int j = lo2;  
  
    while(currentIndex <= hi2) {  
        if(i > hi1) {//we only have a value from the 2nd array to consider  
            storage[currentIndex] = array[j];  
            j++;  
        }  
        else if(j > hi2) {//we only have a value from the 1st array to consider  
            storage[currentIndex] = array[i];  
            i++;  
        }  
        else {//we have to consider a value from both of the arrays  
            if(array[i] < array[j]) {  
                storage[currentIndex] = array[i];  
                i++;  
            }  
            else {  
                storage[currentIndex] = array[j];  
                j++;  
            }  
        }  
        currentIndex++;  
    }  
  
    //copy all the values from the storage array back  
    for(int k = lo1; k <= hi1; k++)  
        array[k] = storage[k];  
}  
  
}
```