

△ Merge Sort

↳ Leverages recursion to perform the sort

- ① if the number of elements to sort is 0 or 1: return
- ② recursively sort 1st and 2nd halves separately
- ③ merge the sorted halves

19	81	42	88	18	14	21	16
----	----	----	----	----	----	----	----

* $O(n \log n)$

19	81	42	88
----	----	----	----

18	14	21	16
----	----	----	----

19	81
----	----

42	88
----	----

18	14
----	----

21	16
----	----

19

81

42

88

18

14

21

16

} individual arrays
are now sorted!

12/05/2011

helper
function →

```
public void mergeSort (int [] a) {  
    int [] tempArray = new int [a.length];  
    mergeSort(a, tempArray, 0, a.length - 1);  
}
```

```
public void mergeSort (int [] a, int temp [], int left, int right)
```

```
{  
    if (left < right) {
```

```
        int center = (left + right) / 2;
```

```
        mergeSort(a, temp, left, center);
```

```
        mergeSort(a, temp, center + 1, right);
```

```
        merge(a, temp, left, center + 1, right);
```

```
    }
```

```
}
```

CODE FOR MERGE IS ON NEXT PAGE