

```
1 // array to sort
2 var array = [9, 2, 5, 6, 4, 3, 7, 10, 1, 8];
3
4 // top-down implementation
5 function mergeSortTopDown(array) {
6     if(array.length < 2) {
7         return array;
8     }
9
10    var middle = Math.floor(array.length / 2);
11    var left = array.slice(0, middle);
12    var right = array.slice(middle);
13
14    return mergeTopDown(mergeSortTopDown(left), mergeSortTopDown(right));
15 }
16 function mergeTopDown(left, right) {
17     var array = [];
18
19     while(left.length && right.length) {
20         if(left[0] < right[0]) {
21             array.push(left.shift());
22         } else {
23             array.push(right.shift());
24         }
25     }
26     return array.concat(left.slice()).concat(right.slice());
27 }
28
29 console.log(mergeSortTopDown(array.slice())); // => [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
30
31 // bottom-up implementation
32 function mergeSortBottomUp(array) {
33     var step = 1;
34     while (step < array.length) {
35         var left = 0;
36         while (left + step < array.length) {
37             mergeBottomUp(array, left, step);
38             left += step * 2;
39         }
40         step *= 2;
41     }
42     return array;
43 }
44 function mergeBottomUp(array, left, step) {
45     var right = left + step;
46     var end = Math.min(left + step * 2 - 1, array.length - 1);
47     var leftMoving = left;
48     var rightMoving = right;
49     var temp = [];
50
51     for (var i = left; i <= end; i++) {
52         if ((array[leftMoving] <= array[rightMoving] || rightMoving > end) &&
53             leftMoving < right) {
54             temp[i] = array[leftMoving];
55             leftMoving++;
56         } else {
57             temp[i] = array[rightMoving];
58             rightMoving++;
59         }
60     }
```

```
61  
62   for (var j = left; j <= end; j++) {  
63     array[j] = temp[j];  
64   }  
65 }  
66  
67 console.log(mergeSortBottomUp(array.slice())); // => [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10  
68 ]
```