

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

```
61     var left = 0;
62     while (left + step < array.length) {
63         countInner++;
64         mergeBottomUp(array, left, step);
65         left += step * 2;
66     }
67     step *= 2;
68 }
69 return array;
70 }
71 function mergeBottomUp(array, left, step) {
72     var right = left + step;
73     var end = Math.min(left + step * 2 - 1, array.length - 1);
74     var leftMoving = left;
75     var rightMoving = right;
76     var temp = [];
77
78     for (var i = left; i <= end; i++) {
79         if ((array[leftMoving] <= array[rightMoving] || rightMoving > end) &&
80             leftMoving < right) {
81             temp[i] = array[leftMoving];
82             leftMoving++;
83         } else {
84             temp[i] = array[rightMoving];
85             rightMoving++;
86         }
87     }
88
89     for (var j = left; j <= end; j++) {
90         countSwap++;
91         array[j] = temp[j];
92     }
93 }
94
95 mergeSortBottomUp(arrayRandom.slice()); // => outer: 4 inner: 9 swap: 36
96 console.log('outer:', countOuter, 'inner:', countInner, 'swap:', countSwap);
97 resetCounters();
98
99 mergeSortBottomUp(arrayOrdered.slice()); // => outer: 4 inner: 9 swap: 36
100 console.log('outer:', countOuter, 'inner:', countInner, 'swap:', countSwap);
101 resetCounters();
102
103 mergeSortBottomUp(arrayReversed.slice()); // => outer: 4 inner: 9 swap: 36
104 console.log('outer:', countOuter, 'inner:', countInner, 'swap:', countSwap);
105 resetCounters();
106
```