```
// array to sort
const array = [9, 2, 5, 6, 4, 3, 7, 10, 1, 8];

// top-down implementation
function mergeSortTopDown(array) {
  if(array.length < 2) {
    return array;
  }

  const middle = Math.floor(array.length / 2);
  const left = array.slice(0, middle);
  const right = array.slice(middle);

  return mergeTopDown(mergeSortTopDown(left), mergeSortTopDown(right));
}
function mergeTopDown(left, right) {
  const array = [];

  while(left.length && right.length) {
    if(left[0] < right[0]) {
      array.push(left.shift());
    } else {
      array.push(right.shift());
    }
  }
  return array.concat(left.slice()).concat(right.slice());
}

console.log(mergeSortTopDown(array.slice())); // => [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]

// bottom-up implementation
function mergeSortBottomUp(array) {
  let step = 1;
  while (step < array.length) {
    let left = 0;
    while (left + step < array.length) {
      mergeBottomUp(array, left, step);
      left += step * 2;
    }
    step *= 2;
  }
  return array;
}
function mergeBottomUp(array, left, step) {
  const right = left + step;
  const end = Math.min(left + step * 2 - 1, array.length - 1);
  let leftMoving = left;
  let rightMoving = right;
  const temp = [];

  for (let i = left; i <= end; i++) {
    if ((array[leftMoving] <= array[rightMoving] || rightMoving > end) &&
        leftMoving < right) {
      temp[i] = array[leftMoving];
      leftMoving++;
    } else {
      temp[i] = array[rightMoving];
      rightMoving++;
    }
  }
```

```
61
62    for (let 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
```