AlgoExpert Quad Layout C# 12px Sublime Monokai 00:00:00

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1

47

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
    public class Program {
     // Best: O(nlog(n)) time | O(log(n)) space
      // Average: O(nlog(n)) time | O(log(n)) space
      // Worst: O(n^2) time | O(log(n)) space
      public static int[] QuickSort(int[] array) {
        QuickSort(array, 0, array.Length - 1);
 9
        return array;
10
11
      public static void QuickSort(int[] array, int startIdx, int endIdx) {
12
13
        if (startIdx >= endIdx) {
14
          return;
15
        int pivotIdx = startIdx;
16
17
        int leftIdx = startIdx + 1;
18
        int rightIdx = endIdx;
19
        while (rightIdx >= leftIdx) {
          if (array[leftIdx] > array[pivotIdx] && array[rightIdx] < array[pivotIdx]) {</pre>
20
21
            swap(leftIdx, rightIdx, array);
22
23
          if (array[leftIdx] <= array[pivotIdx]) {</pre>
           leftIdx += 1;
24
25
26
          if (array[rightIdx] >= array[pivotIdx]) {
27
           rightIdx -= 1;
28
29
30
        swap(pivotIdx, rightIdx, array);
        bool leftSubarrayIsSmaller = rightIdx - 1 - startIdx < endIdx - (rightIdx + 1);</pre>
31
        if (leftSubarrayIsSmaller) {
32
          QuickSort(array, startIdx, rightIdx - 1);
33
34
          QuickSort(array, rightIdx + 1, endIdx);
35
        } else {
          QuickSort(array, rightIdx + 1, endIdx);
36
37
          QuickSort(array, startIdx, rightIdx - 1);
38
39
40
41
      \label{public static void swap(int i, int j, int[] array) {} \\
        int temp = array[j];
        array[j] = array[i];
43
        array[i] = temp;
44
45
46 }
```