AlgoExpert Quad Layout Swift 12px Sublime Monokai 00:00:00

 Prompt
 Scratchpad
 Our Solution(s)
 Video Explanation

Run Code

```
_{\rm 1} // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
 3 class Program {
       // Best: O(nlog(n)) time : O(1) space
        // Average: O(nlog(n)) time : O(1) space
        // Worst: O(nlog(n)) time : O(1) space
        func heapSort(_ array: [Int]) -> [Int] {
           var mutableArray = array
10
           buildHeap(&mutableArray)
11
12
            for index in stride(from: array.count - 1, to: 0, by: -1) {
               swap(0, index, &mutableArray)
13
14
               var startIndex = 0
15
               var endIndex = index - 1
16
17
18
                siftDown(&startIndex, &endIndex, &mutableArray)
19
20
21
           return mutableArray
22
23
24
        func buildHeap(_ array: inout [Int]) {
25
           var firstParentIndex = Double((array.count - 2) / 2)
26
            firstParentIndex = firstParentIndex.rounded(.down)
27
28
            for var currentIndex in (0 \dots Int(firstParentIndex)).reversed() {
29
                var endIndex = array.count - 1
30
                siftDown(&currentIndex, &endIndex, &array)
31
32
33
34
        func siftDown(_ currentIndex: inout Int, _ endIndex: inout Int, _ heap: inout [Int]) {
35
           var firstChildIndex = (currentIndex * 2) + 1
36
37
           while firstChildIndex <= endIndex {</pre>
               var secondChildIndex = -1
38
39
40
                let potentialSecondChildIndex = (currentIndex * 2) + 2
41
42
                if potentialSecondChildIndex <= endIndex {</pre>
43
                    secondChildIndex = potentialSecondChildIndex
44
45
46
               var indexToSwap = -1
47
               if secondChildIndex != -1, heap[secondChildIndex] > heap[firstChildIndex] {
49
                   indexToSwap = secondChildIndex
50
                } else {
                    indexToSwap = firstChildIndex
51
52
53
54
                if heap[indexToSwap] > heap[currentIndex] {
55
                   swap(currentIndex, indexToSwap, &heap)
56
                    currentIndex = indexToSwap
57
                   firstChildIndex = (currentIndex * 2) + 1
58
                } else {
59
                   return
60
61
62
63
64
        func swap(_ firstIndex: Int, _ secondIndex: Int, _ array: inout [Int]) {
65
           let temp = array[firstIndex]
66
67
            array[firstIndex] = array[secondIndex]
68
            array[secondIndex] = temp
69
70 }
```

Solution 1

71