Solution 1

```swift
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

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

        buildHeap(&mutableArray)

        for index in stride(from: array.count - 1, to: 0, by: -1) {
            swap(0, index, &mutableArray)

            var startIndex = 0
            var endIndex = index - 1

            siftDown(&startIndex, &endIndex, &mutableArray)
        }

        return mutableArray
    }

    func buildHeap(_ array: inout [Int]) {
        var firstParentIndex = Double((array.count - 2) / 2)
        firstParentIndex = firstParentIndex.rounded(.down)

        for var currentIndex in (0 ... Int(firstParentIndex)).reversed() {
            var endIndex = array.count - 1
            siftDown(&currentIndex, &endIndex, &array)
        }
    }

    func siftDown(_ currentIndex: inout Int, _ endIndex: inout Int, _ heap: inout [Int]) {
        var firstChildIndex = (currentIndex * 2) + 1

        while firstChildIndex <= endIndex {
            var secondChildIndex = -1

            let potentialSecondChildIndex = (currentIndex * 2) + 2

            if potentialSecondChildIndex <= endIndex {
                secondChildIndex = potentialSecondChildIndex
            }

            var indexToSwap = -1

            if secondChildIndex != -1, heap[secondChildIndex] > heap[firstChildIndex] {
                indexToSwap = secondChildIndex
            } else {
                indexToSwap = firstChildIndex
            }

            if heap[indexToSwap] > heap[currentIndex] {
                swap(currentIndex, indexToSwap, &heap)
                currentIndex = indexToSwap
                firstChildIndex = (currentIndex * 2) + 1
            } else {
                return
            }
        }
    }

    func swap(_ firstIndex: Int, _ secondIndex: Int, _ array: inout [Int]) {
        let temp = array[firstIndex]

        array[firstIndex] = array[secondIndex]
        array[secondIndex] = temp
    }
}

```