AlgoExpert Quad Layout Go 12px Sublime Monokai 00:00:0

Our Solution(s) Run Code

Solution 1

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

package main

type MinHeap []int

func NewMinHeap(array []int) *MinHeap {
  heap := MinHeap(array)
  ptr := &heap
  ptr.BuildHeap(array)
  return ptr
}

// O(n) time | O(1) space
func (h *MinHeap) BuildHeap(array []int) {
  first := (len(array) - 2) / 2
  for currentIndex := first + 1; currentIndex >= 0; currentIndex-- {
    h.siftDown(currentIndex, len(array)-1)
  }
}

// O(log(n)) time | O(1) space
func (h *MinHeap) siftDown(currentIndex, endIndex int) {
  childOneIdx := currentIndex*2 + 1
  for childOneIdx <= endIndex {
    childTwoIdx := -1
    if currentIndex*2+2 <= endIndex {
      childTwoIdx = currentIndex*2 + 2
    }
    indexToSwap := childOneIdx
    if childTwoIdx > -1 && (*h)[childTwoIdx] < (*h)[childOneIdx] {
      indexToSwap = childTwoIdx
    }
    if (*h)[indexToSwap] < (*h)[currentIndex] {
      h.swap(currentIndex, indexToSwap)
      currentIndex = indexToSwap
      childOneIdx = currentIndex*2 + 1
    } else {
      return
    }
  }
}

// O(log(n)) time | O(1) space
func (h *MinHeap) siftUp() {
  currentIndex := h.length() - 1
  parentIndex := (currentIndex - 1) / 2
  for currentIndex > 0 {
    current, parent := (*h)[currentIndex], (*h)[parentIndex]
    if current < parent {
      h.swap(currentIndex, parentIndex)
      currentIndex = parentIndex
      parentIndex = (currentIndex - 1) / 2
    } else {
      return
    }
  }
}

// O(1) time | O(1) space
func (h MinHeap) Peek() int {
  if len(h) == 0 {
    return -1
  }
  return h[0]
}

// O(log(n)) time | O(1) space
func (h *MinHeap) Remove() int {
  l := h.length()
  h.swap(0, l-1)
  peeked := (*h)[l-1]
  *h = (*h)[0 : l-1]
  h.siftDown(0, l-1)
  return peeked
}

// O(log(n)) time | O(1) space
func (h *MinHeap) Insert(value int) {
  *h = append(*h, value)
  h.siftUp()
}

func (h MinHeap) swap(i, j int) {
  h[i], h[j] = h[j], h[i]
}

func (h MinHeap) length() int {
  return len(h)
}
```

Your Solutions Run Code

Solution 1 **Solution 2** **Solution 3**

```go
package main

// Do not edit the class below except for the buildHeap,
// siftDown, siftUp, peek, remove, and insert methods.
// Feel free to add new properties and methods to the class.
type MinHeap []int

func NewMinHeap(array []int) *MinHeap {
  // Do not edit the lines below.
  heap := MinHeap(array)
  ptr := &heap
  ptr.BuildHeap(array)
  return ptr
}

func (h *MinHeap) BuildHeap(array []int) {
  // Write your code here.
}

func (h *MinHeap) siftDown(currentIndex, endIndex int) {
  // Write your code here.
}

func (h *MinHeap) siftUp() {
  // Write your code here.
}

func (h MinHeap) Peek() int {
  // Write your code here.
  return -1
}

func (h *MinHeap) Remove() int {
  // Write your code here.
  return -1
}

func (h *MinHeap) Insert(value int) {
  // Write your code here.
}

```

Custom Output **Raw Output** Submit Code

91

Run or submit code when you're ready.