Tópicos Avançados em Estrutura de Dados

Atividade 16

Bruna Galastri Guedes	18.00189-0
Daniel Ughini Xavier	18.00022-3
Rodolfo Cochi Bezerra	18.00202-0
Vítor Martin Simoni	18.00050-9
Leonardo Cury Haddad	18.00442-3
Leonardo de Barros Rodrigues	18.02401-7

11/08/2020

Algoritmo de HeapSort implementado em Go

```go
package main

import "fmt"

func main() {

        var myArray = []int{9, 83, 8, 100, 19, 43, 1, 42, 78, 75, 93, 3}

        heapsort(myArray)
        fmt.Println(myArray)

}

func heapsort(arr []int) {
        n := len(arr)
        buildmaxheap(arr, n)
        for n > 0 {
            //Swap
                aux := arr[0]
                arr[0] = arr[n-1]
                arr[n-1] = aux
                n -= 1
                heapify(0, arr, n)
        }
}

func buildmaxheap(arr []int, n int) {
        for i := n/2 - 1; i >= 0; i-- {
                heapify(i, arr, n)
        }
}

func heapify(index int, arr []int, n int) {
        largest := index
        left := index*2 + 1
        right := index*2 + 2
        if left < len && arr[left] > arr[largest] {
                largest = left
        }
        if right < n && arr[right] > arr[largest] {
                largest = right
        }
        if largest != index {
                //Swap
                aux := arr[largest]
                arr[largest] = arr[index]
                arr[index] = aux

                heapify(largest, arr, n)
        }
}

//Saida do programa: [1 3 8 9 19 42 43 75 78 83 93 100]
```