

# **Tópicos Avançados em Estrutura de Dados**

## **Atividade 16 - Heapsort**

Guilherme Samuel de Souza Barbosa	19.00012-0
Guilherme Cury Galli	19.00374-9
Gustavo Consoleti Ramirez de Souza	19.00715-9
Igor Eiki Ferreira Kubota	19.02466-5

23 de agosto de 2021

## Questão 1

A implementação do algoritmo HeapSort em Python foi feita da seguinte forma:

# Código em Python que implementa o algoritmo de ordenação HeapSort

```
def heapify(arr, n, i): #n é o tamanho do heap
    larg = i # inicializa o mais largo como root
    esq = 2 * i + 1 # esquerda = 2*i + 1
    dir = 2 * i + 2 # direita = 2*i + 2

    # Veja se o filho esquerdo do root existe e se é
    # maior que o root
    if esq < n and arr[larg] < arr[esq]:
        larg = esq

    # Veja se o filho direito do root existe e se é
    # maior que o root
    if dir < n and arr[larg] < arr[dir]:
        larg = dir

    # Se necessario, modificar o root
    if larg != i:
        arr[i], arr[larg] = arr[larg], arr[i] # swap

    # Heapify no root.
    heapify(arr, n, larg)

# A função para ordenar um array de determinado tamanho

def heapSort(arr):
    n = len(arr)

    # Cria um maxheap.
    for i in range(n//2 - 1, -1, -1):
        heapify(arr, n, i)

    for i in range(n-1, 0, -1):
        arr[i], arr[0] = arr[0], arr[i]
        heapify(arr, i, 0)

arr = [12, 11, 13, 5, 6, 7]
heapSort(arr)
n = len(arr)
print("Array organizado: ")
for i in range(n):
    print("%d" % arr[i], end=" ")
```