

## Algoritmos e Estrutura de Dados II (ESTCMP011)

2do PERIODO 2018

Prof. Dr. Luis Cuevas Rodríguez

Trabalho pratica: Heap Sort

Equipes de quatro pessoas máximas.

Objetivo: Analisar o comportamento do algoritmo Heap Sort e comparar ele com outros algoritmos de ordenação estudados.

Tarefas

1. Programar o algoritmo de ordenação Heap Sort (tem um exemplo no final do arquivo)

2. Testar ele com nove listas de elementos.

a) Lista ordenada crescente

a) Lista ordenada decrescente

c) Lista desordenada aleatoriamente.

Usar para cada tipo de listas, outras três listas com, 50, 500 e 1000 elementos.

3. Selecionar outro algoritmo de ordenação estudado e testar ele com as mesmas listas.

4. Analisar os resultados por: Tempo de execução, número de comparações feitas e número de movimentações feitas.

5. Fazer um relatório com os resultados.

### Exemplo do Código

```
#define TAM 5  
int tamHeap;
```

```

int main()
{
    int lista[TAM]={5,9,1,6,8};
    imprime_vetor(lista);
    heapSort(lista,TAM);
    cout << endl;
    imprime_vetor(lista);
    return 0;
}

void heapify(int arr[], int i)
{
    int l = 2*i + 1;
    int r = 2*i + 2;
    int m = i;
    int temp;
    if (l < tamHeap && arr[l] > arr[m])
        m = l;
    if (r < tamHeap && arr[r] > arr[m])
        m = r;
    if (m != i)
    {
        temp=arr[i];
        arr[i]=arr[m];
        arr[m]=temp;
        heapify(arr, m);
    }
}

void constHeap(int arr[], int n)
{
    tamHeap=n;
    for (int i = n / 2 - 1; i >= 0; i--)
        heapify(arr,i);
}

void heapSort(int arr[], int n)
{
    int temp;
    constHeap(arr, n);
    for (int i=n-1; i>=0; i--)
    {
        temp=arr[0];
        arr[0]=arr[i];
        arr[i]=temp;
        tamHeap--;
        heapify(arr,0);
    }
}

```