

Universidade De Évora  
Departamento de Informática  
Estruturas de Dados e Algoritmos I  
Ano letivo 2018 - 2019

# Implementação de Algoritmos de Ordenação

Alunos: Luís Ressonha - 35003  
Rúben Teimas - 39868

Docente: Lígia Ferreira

Janeiro 2019

## Introdução

O trabalho consiste na implementação de 2 algoritmos de ordenação distintos e de complexidade distinta. Os algoritmos escolhidos poderão ser algum(s), dos algoritmos dados/estudados ou não.

A implementação apresentada, deve receber do standard input uma sequência de valores (comparáveis) e realizar a sua ordenação.

Nós escolhemos implementar *InsertionSort* e *HeapSort*

# 1 Descrição e Funcionamento do Trabalho

## 1.1 insertionSort

No algoritmo InsertionSort comparamos o index com o index-1 e guardamos o index-element se o segundo index for mais pequeno que o primeiro. Fazemos este processo recursivamente até estar bem colocado.

Quando estiver colocado, colocamos o primeiro index-element no index atual e continuamos este processo até o array estar todo ordenado.

## 1.2 HeapSort

Calculamos o index do ultimo heap da direita, comparamos os filhos com os pais, caso o filho seja maior do que o pai estes trocam, seguidamente passamos para a esquerda (tudo isto recursivamente). Ao estar construída a max-heap, trocamos o maior valor com menor e voltamos a fazer heapify(primeiro método). Tudo isto até o array estar ordenado.

## Conclusão

Como foi dito na introdução deste relatório, o trabalho consiste na implementação de 2 algoritmos de ordenação distintos e de complexidade distinta. Os algoritmos escolhidos foram o *InsertionSort* e o *HeapSort*

Na realização deste trabalho não foram atingidos todos os objetivos. Ao utilizar o método *le-array()* não nos foi possível compara strings, contudo ao criar um array na *main*, não usando este método, conseguimos ordenar um array *comparable*. Concluímos por isso que o problema estava no método *le-array()*.