



UNIVERSIDADE DE COIMBRA  
FACULDADE DE CIÊNCIAS E TECNOLOGIA

**Departamento de Engenharia  
Informática**

## **Projeto #4 Algoritmos e Estruturas de Dados**

**2021-2022 – 1º Semestre**

**Submissão de relatório (InforEstudante) e código  
(Mooshak):**

4.1 21 de novembro 23:59    4.2 28 de novembro 23:59  
4.3 12 de dezembro 23:59    4.4 19 de dezembro 23:59

**Anotações:** Em anexo ao enunciado do projeto são disponibilizados quatro templates do relatório para cada uma das semanas de duração do projeto.

*É incentivado que os alunos discutam ideias e questões relativas ao trabalho proposto, mas é entendido que quer a reflexão final sobre os resultados obtidos, quer o código desenvolvido, são da autoria de cada estudante. Procedimentos contrários ao que é dito acima, nomeadamente cópia de código desenvolvido por colegas ou obtido da net é entendido como fraude. Para além de a fraude denotar uma grave falta de ética e constituir um comportamento não admissível num estudante do ensino superior e futuro profissional licenciado, esta prejudica definitivamente o processo de aprendizagem do infrator.*

### **Objetivos:**

Com o desenvolvimento deste projeto pretende-se que o aluno consolide os conhecimentos sobre os algoritmos de ordenamento estudados em Algoritmos e Estruturas de Dados com foco na: (1) programação dos algoritmos; (2) vantagens e desvantagens de cada uma, nomeadamente no que diz respeito a complexidade temporal e espacial; (3) análise teórica e empírica da complexidade temporal.

---

## **Base comum a todos os subprojetos**

---

O ordenamento será feito sobre sequências de inteiros com valores entre 0 e 10,000,000. O resultado do ordenamento será a sequência de entrada ordenada de forma crescente.

### **Entrada:**

A primeira linha indica o número de inteiros a ordenar (N). De seguida existem N linhas, com um inteiro por linha.

### **Saída:**

O output terá N linhas, com um inteiro por linha.

### **Exemplo:**

**Entrada:**

4  
17  
21  
5  
51

**Saída:**

5  
17  
21  
51

Os diferentes tipos de sequências a testar serão os seguintes:

- 1) SEQ\_ALEATORIA: sequência aleatória.
- 2) SEQ\_ORDENADA\_DECRESCENTE: sequência ordenada de forma decrescente.
- 3) SEQ\_QUASE\_ORDENADA\_1% e SEQ\_QUASE\_ORDENADA\_5%: sequência quase ordenada com x% de trocas a criarem desordenamento. Para criarem este tipo de sequência podem começar com uma sequência ordenada e depois sorteiam x% de trocas aleatórias.

Estas quatro sequências serão usadas em todos os subprojetos com os seus nomes respetivos: SEQ\_ALEATORIA, SEQ\_ORDENADA\_DECRESCENTE, SEQ\_QUASE\_ORDENADA\_1% e SEQ\_QUASE\_ORDENADA\_5%.

Os relatórios dos subprojetos 4.1, 4.2 e 4.3 serão realizados com base nos templates disponibilizados e devem ter em conta:

- Gráficos com as medições e resultado da regressão para cada algoritmo e para cada sequência
- Reflexão crítica sobre o resultado da regressão e possíveis *outliers*
- Análise de complexidade com base nos resultados empíricos obtidos

O relatório do subprojeto 4.4 será realizado com base no template disponibilizado e deve ter em conta:

- Gráficos com as medições para cada algoritmo e para cada sequência
- Análise de complexidade com base nos resultados empíricos obtidos

## **Subprojeto 4.1: Bubble sort**

Implementação do Bubble sort e estudo empírico do algoritmo para todos os tipos de sequência descritos na secção inicial. No Mooshak devem submeter o Bubble sort no problema A.

## **Subprojeto 4.2: Insertion sort**

Implementação do Insertion sort e estudo empírico do algoritmo para todos os tipos de sequência descritos na secção inicial. No Mooshak devem submeter o Insertion sort no problema B.

## **Subprojeto 4.3: Quicksort**

Implementação do Quicksort com exploração de diferentes métodos de seleção do pivot.

Método A: o pivot é escolhido como sendo o primeiro elemento do subconjunto considerado.

Método B: o pivot é escolhido como sendo um elemento aleatório do subconjunto considerado.

Método C: o pivot é escolhido como sendo a mediana do primeiro, último, e elemento do meio do subconjunto considerado.

Devem analisar empiricamente estas variantes para os quatro tipos de sequência descritos na secção inicial. No Mooshak devem submeter o Quicksort usando o método A no problema C, o Quicksort usando o método B no problema D, e o Quicksort usando o método C no problema E.

## **Subprojeto 4.4: Análise comparativa dos algoritmos de ordenamento dos subprojetos anteriores**

Neste subprojeto devem ser comparadas as variantes abordadas nos subprojetos anteriores:

- Bubble sort
- Insertion sort
- Quicksort com os 3 métodos de seleção do pivot

Devem reutilizar os resultados obtidos nos subprojetos anteriores e fazer uma análise comparativa. Adicionalmente, devem apresentar uma discussão sobre a estabilidade de cada algoritmo de ordenamento e, nos casos em que o algoritmo não é estável, propor alterações ao algoritmo de forma a garantir a sua estabilidade. Finalmente, devem discutir a possível criação de um algoritmo

híbrido com base em 2 ou 3 destes algoritmos. Existirão vantagens gerais, vantagens para alguns tipos de sequência, ou nunca será vantajoso? Justifique.

# Relatório Projeto 4.1 AED 2021/2022

Nome:

Nº Estudante:

TP (inscrição):

*Login no Mooshak:*

Registrar os tempos computacionais do Bubble sort para os diferentes tipos de sequências. Os tamanhos das sequências (N) devem ser: 25000, 50000, 75000, 100000, 125000. Só deve ser contabilizado o tempo de ordenamento. Exclui-se o tempo de leitura do input e de impressão dos resultados. Devem apresentar e discutir as regressões para cada tipo de sequência.

**Gráfico para SEQ\_ALEATORIA**

**Gráfico para SEQ\_ORDENADA\_DECRESCENTE**

**Gráfico para SEQ\_QUASE\_ORDENADA\_1%**

**Gráfico para SEQ\_QUASE\_ORDENADA\_5%**

**Análise dos resultados:**

## Relatório Projeto 4.2 AED 2021/2022

Nome:

Nº Estudante:

TP (inscrição):

*Login no Mooshak:*

Registrar os tempos computacionais do Insertion sort para os diferentes tipos de sequências. Os tamanhos das sequências (N) devem ser: 25000, 50000, 75000, 100000, 125000. Só deve ser contabilizado o tempo de ordenamento. Exclui-se o tempo de leitura do input e de impressão dos resultados. Devem apresentar e discutir as regressões para cada tipo de sequência.

**Gráfico para SEQ\_ALEATORIA**

**Gráfico para SEQ\_ORDENADA\_DECRESCENTE**

**Gráfico para SEQ\_QUASE\_ORDENADA\_1%**

**Gráfico para SEQ\_QUASE\_ORDENADA\_5%**

**Análise dos resultados:**

## Relatório Projeto 4.3 AED 2021/2022

Nome:

Nº Estudante:

TP (inscrição):

*Login no Mooshak:*

Registrar os tempos computacionais do Quicksort para os 3 métodos de seleção do pivot e para os diferentes tipos de sequências. Os tamanhos das sequências (N) devem ser: 25000, 50000, 75000, 100000, 125000. Só deve ser contabilizado o tempo de ordenamento. Exclui-se o tempo de leitura do input e de impressão dos resultados. Devem apresentar e discutir as regressões para cada variante e para cada tipo de sequência.

**Gráfico para SEQ\_ALEATORIA**

**Gráfico para SEQ\_ORDENADA\_DECRESCENTE**

**Gráfico para SEQ\_QUASE\_ORDENADA\_1%**

**Gráfico para SEQ\_QUASE\_ORDENADA\_5%**

**Análise dos resultados:**

# Relatório Projeto 4.4 AED 2021/2022

Nome:

Nº Estudante:

TP (inscrição):

*Login no Mooshak:*

Registrar os tempos computacionais das variantes em consideração para os diferentes tipos de sequências. Os tamanhos das sequências (N) devem ser: 25000, 50000, 75000, 100000, 125000. Só deve ser contabilizado o tempo de ordenamento. Exclui-se o tempo de leitura do input e de impressão dos resultados.

**Gráfico para SEQ\_ALEATORIA**

**Gráfico para SEQ\_ORDENADA\_DECRESCENTE**

**Gráfico para SEQ\_QUASE\_ORDENADA\_1%**

**Gráfico para SEQ\_QUASE\_ORDENADA\_5%**

**Análise dos resultados + discussão sobre a estabilidade de cada algoritmo de ordenamento e, nos casos em que o algoritmo não é estável, propor alterações ao algoritmo de forma a garantir a sua estabilidade + discussão sobre a possível criação de um algoritmo híbrido com base em 2 ou 3 destes algoritmos. Existirão vantagens gerais, vantagens para alguns tipos de sequência, ou nunca será vantajoso? Justifique:**