

 <p>UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA</p> <p>Departamento de Engenharia Informática</p>	<p>Projeto #3 Algoritmos e Estruturas de Dados</p> <p>2021-2022 – 1º Semestre</p> <p>Submissão de relatório (InforEstudante) e código (Mooshak):</p> <table> <tr> <td>3.1 18 de outubro 23:59</td><td>3.2 24 de outubro 23:59</td></tr> <tr> <td>3.3 7 de novembro 23:59</td><td>3.4 14 de novembro 23:59</td></tr> </table>	3.1 18 de outubro 23:59	3.2 24 de outubro 23:59	3.3 7 de novembro 23:59	3.4 14 de novembro 23:59
3.1 18 de outubro 23:59	3.2 24 de outubro 23:59				
3.3 7 de novembro 23:59	3.4 14 de novembro 23:59				
<p>Anotações: Em anexo ao enunciado do projeto são disponibilizados quatro templates do relatório para cada uma das semanas de duração do projeto.</p> <p><i>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.</i></p>					

Objetivos:

Com o desenvolvimento deste projeto pretende-se que o aluno consolide os conhecimentos sobre as estruturas de dados estudadas 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 à complexidade temporal; (3) análise teórica e empírica da complexidade temporal.

Base comum a todos os subprojetos: sistema de gestão de vacinação

Pretende-se desenvolver um programa que guarda registos dos utentes da uma rede nacional de vacinação. Cada registo compreende o **número de utente** seguido um conjunto de dados associados de vacinação. Para cada utente é inicialmente criado um registo que depois vai ser acedido com bastante frequência (muito mais consultas ao sistema do que inserções, seja para incluir novas vacinações seja para verificar se o utente tem uma determinada vacina em dia).

As operações (comandos) possíveis sobre o Sistema de Vacinação são:

1. ACRESCENTA <numUtente> <vacina> <data limite>

Se ainda não existe um registo com < numUtente > cria esse registo. Se já existe verifica se a <vacina> já existe e nesse caso atualiza <data limite>, se vacina ainda não existe acrescenta <vacina> e <data limite>. A data deve ser inserida no formato ddmmyyyy.

Terminada a inserção deve imprimir uma linha com “NOVO UTENTE CRIADO” se ainda não existir o < numUtente > no sistema; “NOVA VACINA INSERIDA” se já existe *numUtente* mas ainda não tem informação sobre essa vacina; “VACINA ATUALIZADA” se vacina já existe, caso em que só atualiza data.

2. CONSULTA < numUtente >

Mostra todas as vacinas e datas limite associadas a < numUtente >. Vacinas apresentadas por ordem alfabética. Termina com uma linha com a palavra “FIM”.

Se < numUtente > não está presente no sistema devolve uma linha com “NAO ENCONTRADO”.

3. LISTAGEM

Faz a listagem de todos os < numUtente > guardados e respetivas <vacinas> e <data limite> seguida de linha com a palavra “FIM”. < numUtente > e <vacinas> por ordem crescente.

4. APAGA

Elimina todos os registos no sistema, de seguida imprime a linha com “LISTAGEM DE NOMES APAGADA”

5. FIM

Termina a sequência de comandos com uma linha com a palavra “FIM”:

Entrada:

Como entrada o programa recebe todos os comandos a realizar, um por linha.

Saída:

Os resultados para todos os comandos realizados, tal como especificado acima.

Exemplos:

Entrada

```
APAGA
ACRESCENTA 12340 polio 20022025
ACRESCENTA 56700 covid 10082023
ACRESCENTA 88100 tuberculose 15122030
CONSULTA 56700
ACRESCENTA 56700 covid 10102024
ACRESCENTA 56700 papeira 12112026
LISTAGEM
APAGA
FIM
```

Saída

LISTAGEM DE NOMES APAGADA
NOVO UTENTE CRIADO
NOVO UTENTE CRIADO
NOVO UTENTE CRIADO
56700 covid 10082023
FIM
VACINA ATUALIZADA
NOVA VACINA INSERIDA
12340 polio 20022025
56700 covid 10102024 papeira 12112026
88100 tuberculose 15122030
FIM
LISTAGEM DE NOMES APAGADA

Entrada

ACRESCENTA 23456 polio 20032023
CONSULTA 23456
APAGA
CONSULTA 23456
FIM

Saída

NOVO UTENTE CRIADO
23456 polio 20032023
FIM
LISTAGEM DE NOMES APAGADA
NAO ENCONTRADO

Entrada

ACRESCENTA 234 polio 20122025
ACRESCENTA 123 zeneca 12122030
ACRESCENTA 234 abacus 12032026
ACRESCENTA 123 covid 2122026
ACRESCENTA 123 abacus 2122032
CONSULTA 123
LISTAGEM
APAGA
CONSULTA 123
FIM

Saída

NOVO UTENTE CRIADO

NOVO UTENTE CRIADO

NOVA VACINA INSERIDA

NOVA VACINA INSERIDA

NOVA VACINA INSERIDA

123 abacus 2122032 covid 2122026 zeneca 12122030

FIM

123 abacus 2122032 covid 2122026 zeneca 12122030

234 abacus 12032026 polio 20122025

FIM

LISTAGEM DE NOMES APAGADA

NAO ENCONTRADO

Nos estudos experimentais devem ser explorados os seguintes valores para N: 10000, 100000, 1000000, 10000000, e devem ser considerados dois cenários:

- Cenário 1: 10% de inserções e 90% de pesquisas
- Cenário 2: 90% de inserções e 10% de pesquisas

As inserções são baseadas na execução do comando **ACRESCENTA**, enquanto que as pesquisas são baseadas na execução do comando **CONSULTA**.

Os relatórios dos subprojetos 3.1, 3.2 e 3.3 serão realizados com base nos templates disponibilizados e devem ter em conta:

- Tabelas concisas com as medições efetuadas para cada cenário
- Gráficos com as medições e resultado da regressão para cada solução desenvolvida e para cada cenário
- Reflexão crítica sobre o resultado da regressão e possíveis *outliers*
- Análise de complexidade com base nos resultados empíricos obtidos

O relatório do subprojeto 3.4 será realizado com base no template disponibilizado e deve ter em conta:

- Tabelas concisas com as medições efetuadas para cada cenário
- Gráficos com as medições para cada solução desenvolvida e para cada cenário
- Análise de complexidade com base nos resultados empíricos obtidos

Subprojeto 3.1: Pesquisa binária sobre arrays

Implementação do algoritmo de pesquisa binária sobre arrays e comparação com a pesquisa linear. A pesquisa binária só se aplica sobre arrays ordenados. Como tal, devem garantir que a operação de inserção é realizada de forma ordenada de modo a garantir que a pesquisa binária funciona como desejado. No Mooshak devem submeter a pesquisa linear no problema A e a pesquisa binária no problema B.

Subprojeto 3.2: Árvore binária de pesquisa

Implementação de uma árvore binária de pesquisa e comparação com a pesquisa linear. No Mooshak devem submeter a árvore binária de pesquisa no problema C.

Subprojeto 3.3: Árvore AVL

Implementação de uma árvore AVL e comparação com a pesquisa linear. No Mooshak devem submeter a árvore AVL no problema D.

Subprojeto 3.4: Análise comparativa das variantes dos subprojetos anteriores

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

- Pesquisa linear
- Pesquisa binária
- Árvore de pesquisa binária
- Árvore AVL

Devem reutilizar os resultados obtidos dos subprojetos anteriores e fazer uma análise comparativa. Adicionalmente, devem apresentar uma discussão sobre como se comportariam estas variantes na operação de remover um elemento do conjunto.

Relatório Projeto 3.1 AED 2021/2022

Nome:

Nº Estudante:

TP (inscrição):

Login no Mooshak:

Registrar os tempos computacionais da pesquisa linear e da pesquisa binária para os dois cenários. Os tamanhos das sequências (N) devem ser: 10000, 100000, 1000000, 10000000, 100000000. Só devem ser contabilizados os tempos das operações de inserção e pesquisa. Exclui-se o tempo de leitura do input e de impressão dos resultados. Devem apresentar e discutir as regressões para as duas variantes e para cada cenário.

Tabela para o cenário 1

Tabela para o cenário 2

Gráfico para o cenário 1

Gráfico para o cenário 2

Análise dos resultados:

Relatório Projeto 3.2 AED 2021/2022

Nome:

Nº Estudante:

TP (inscrição):

Login no Mooshak:

Registrar os tempos computacionais da pesquisa linear e da árvore binária de pesquisa para os dois cenários. Os tamanhos das sequências (N) devem ser: 10000, 100000, 1000000, 10000000, 100000000. Só devem ser contabilizados os tempos das operações de inserção e pesquisa. Exclui-se o tempo de leitura do input e de impressão dos resultados. Devem apresentar e discutir as regressões para as duas variantes e para cada cenário.

Tabela para o cenário 1

Tabela para o cenário 2

Gráfico para o cenário 1

Gráfico para o cenário 2

Análise dos resultados:

Relatório Projeto 3.3 AED 2021/2022

Nome:

Nº Estudante:

TP (inscrição):

Login no Mooshak:

Registrar os tempos computacionais da pesquisa linear e da árvore AVL para os dois cenários. Os tamanhos das sequências (N) devem ser: 10000, 100000, 1000000, 10000000, 100000000. Só devem ser contabilizados os tempos das operações de inserção e pesquisa. Exclui-se o tempo de leitura do input e de impressão dos resultados. Devem apresentar e discutir as regressões para as duas variantes e para cada cenário.

Tabela para o cenário 1

Tabela para o cenário 2

Gráfico para o cenário 1

Gráfico para o cenário 2

Análise dos resultados:

Relatório Projeto 3.4 AED 2021/2022

Nome:

Nº Estudante:

TP (inscrição):

Login no Mooshak:

Registrar os tempos computacionais das variantes exploradas nos subprojetos anteriores para os dois cenários. Os tamanhos das sequências (N) devem ser: 10000, 100000, 1000000, 10000000, 100000000. Só devem ser contabilizados os tempos das operações de inserção e pesquisa. Exclui-se o tempo de leitura do input e de impressão dos resultados. Não devem apresentar regressões.

Tabela para o cenário 1

Tabela para o cenário 2

Gráfico para o cenário 1

Gráfico para o cenário 2

Análise dos resultados e, adicionalmente, discussão sobre como se comportariam estas variantes na operação de remover um elemento do conjunto: