UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Departamento de Engenharia Informática

Projeto #3
Algoritmos e Estruturas de Dados

2021-2022 – 2º Semestre

VERSÃO 1.3

Submissão relatório (InforEstudante) e Mooshak:
3.1 13 de março 23:59 3.2 20 de março 23:59
3.3 3 de abril 23:59 3.4 10 de abril 23:59

Anotações: *este projeto foi preparado para ser resolvido parcialmente no espaço das quatro sessões práticas. Vão ser disponibilizados quatro formulários dos relatórios parciais 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 a estruturas de dados estudadas em Algoritmos e Estruturas de Dados com foco na (1) programação das estruturas (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.

Sub-projeto 3.1: Sistema de cálculo de valorização de categorias de NFTs

Num sistema de catálogo de NFTs, os artigos estão organizados em categorias. Estas categorias têm uma estrutura hierárquica, podendo incluir várias subcategorias. Cada categoria de artigos no catálogo tem um valor calculado de acordo com as transações nessa categoria. Pretende-se desenvolver um programa que calcula o valor acumulado para cada categoria com base nas transações das suas subcategorias do catálogo de artigos. O programa recebe um comando com uma categoria, o seu valor direto e o número de nós filho: <categoria> <valor> <número de filhos>. De acordo com número de filhos, o programa recebe a mesma instrução para cada filho.

A saída do programa será a impressão de cada nível de categorias, da mais genérica para a mais específica, de acordo com a ordem de inserção, e o valor acumulado de cada categoria da seguinte forma: <categoria>(<valor>).

O valor <categoria> é sempre apenas uma palavra. O <valor> é um número inteiro.

Para desenvolver este subprojecto:

- **A.** Escolher a estrutura de dados mais adequada
- **B.** Implementar a estrutura de dados e desenvolver o programa de acordo com a entrada/saída pretendida
- **C.** Analisar a complexidade temporal

Exemplos:

Entrada

Todos 0 3
Arte 5 2
Classica 1000 0
Fotografia 50 0
Livros 100 0
Musica 0 3
Rock 20 1
SoftRock 5 0
Pop 20 0
Country 20 0

Saída

Todos(1220)
Arte(1055) Livros(100) Musica(65)
Classica(1000) Fotografia(50) Rock(25) Pop(20) Country(20)
SoftRock(5)

Compreende a elaboração do Relatório 3.1 (formulário fornecido) e vai incidir sobre a análise de complexidade temporal do programa desenvolvido.

O relatório a realizar com base no formulário disponibilizado deve ter em conta:

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

Relatório (a submeter no InforEstudante)

Sub-projeto 3.2: Sistema de catálogo de NFTs

Pretende-se desenvolver um programa que guarda e apresenta um catálogo de artigos NFT disponíveis para leilão. Visto que alguns destes artigos se tornam virais nas redes sociais, o acesso a estes artigos pode ser bastante assimétrico – 90% dos acessos são feitos a 5% dos artigos.

As operações (comandos) possíveis sobre o sistema de catálogo de NFTs são:

1. **ARTIGO** <nome> <hash> <valor da oferta atual>

Se ainda não existe um registo com <nome> cria esse registo. Se já existe devolve a mensagem "ARTIGO JA EXISTENTE" e não faz nada no registo de artigos. Se insere um novo artigo termina com uma linha com a frase "NOVO ARTIGO INSERIDO"

O <nome> é composto por uma única palavra e a <hash> por 6 caracteres alfanuméricos. O <valor da oferta atual> é do tipo inteiro.

2. **OFERTA** <nome> <valor da oferta atual>

Atualiza o valor da oferta atual anteriormente guardado. Termina com uma linha com a frase "OFERTA ATUALIZADA". SE <nome> não existe no sistema termina com a frase "ARTIGO NAO REGISTADO".

3. **CONSULTA** <nome>

Mostra os dados do artigo, a saber <nome> <hash> <valor da oferta atual> a que se segue a linha "FIM".

Se <nome> não existe no sistema termina com uma linha com a frase "ARTIGO NAO REGISTADO".

4. **LISTAGEM**

Faz a listagem de todos os <nomes> guardados e respetivas <hash> <valor da oferta atual>, um por linha. A listagem deve apresentar os artigos por ordem alfabética. Termina com uma linha com a palavra "FIM".

<nomes> apresentados por ordem alfabética.

5. **APAGA**

Elimina todos os registos no sistema. Imprime uma linha com "CATALOGO APAGADO".

6. **FIM**

Termina a sequência de comandos.

Para desenvolver este subprojecto:

- **A.** Escolher a estrutura de dados principal mais adequada.
- **B.** Implementar a estrutura de dados e desenvolver o programa de acordo com a entrada/saída pretendida.
- **C.** Analisar a complexidade temporal.

Entrada:
Como entrada o programa recebe de uma só vez todos os comandos a realizar, um por linha.

Saída: Os resultados para todos os comandos ativados, tal como especificado acima.

Exemplos:

Entrada

ARTIGO Pintura1 abcdef 100

ARTIGO Pintura1 abcdef 150

OFERTA Pintura1 110

CONSULTA Pintura1

CONSULTA Pintura2

ARTIGO Pintura2 abcde3 150

LISTAGEM

APAGA

FIM

Saída

NOVO ARTIGO INSERIDO

ARTIGO JA EXISTENTE

OFERTA ATUALIZADA

Pintura1 abcdef 110

FIM

ARTIGO NAO REGISTADO

NOVO ARTIGO INSERIDO

Pintura1 abcdef 110

Pintura2 abcde3 150

FIM

CATALOGO APAGADO

Compreende a elaboração do Relatório 3.2 (formulário fornecido) e vai incidir sobre a análise de complexidade temporal do programa desenvolvido.

O relatório a realizar com base no formulário disponibilizado deve ter em conta:

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

Relatório (a submeter no InforEstudante)

Sub-projeto 3.3: Sistema de utilizadores da plataforma de leilões de NFTs

Pretende-se desenvolver um programa que guarda registos dos utilizadores de um website de leilões de NFTs. Cada registo compreende o username seguido de um conjunto de dados associados ao mesmo. Para cada utilizador é inicialmente criado um registo que depois vai ser acedido com bastante frequência (muito mais consultas ao sistema do que inserções, seja para incluir novos dados de faturação seja para verificar a validade dos dados de faturação).

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

1. **ACRESCENTA** <username> <número de cartão de crédito> <data de expiração>

Se ainda não existe um registo com <username> cria esse registo. Se já existe verifica se o <username> já existe e nesse caso atualiza <data limite>, se o cartão ainda não existe acrescenta <cartão> e <data limite>. A data deve ser inserida no formato yymm.

Terminada a inserção deve imprimir uma linha com "NOVO UTILIZADOR CRIADO" se ainda não existir o <username> no sistema; "NOVO CARTAO INSERIDO" se já existe username mas ainda não tem informação sobre esse cartão; "CARTAO ATUALIZADO" se cartão já existe, caso em que só atualiza data.

2. **CONSULTA** <username>

Mostra todos os cartões e datas de expiração associadas a <username>. Cartões apresentados por ordem alfabética. Termina com uma linha com a palavra "FIM".

Se <username> não está presente no sistema devolve uma linha com "NAO ENCONTRADO".

3. **LISTAGEM**

Faz a listagem de todos os <username> guardados e respetivos <cartões> e <data de expiração> seguida de linha com a palavra "FIM". <username> e <cartão> por ordem alfabética.

4. **APAGA**

Elimina todos os registos no sistema, de seguida imprime a linha com "LISTAGEM APAGADA".

5. **FIM**

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

Para desenvolver este subprojecto:

- **A.** Escolher a estrutura de dados principal mais adequada.
- **B.** Implementar a estrutura de dados e desenvolver o programa de acordo com a entrada/saída pretendida.
- **C.** Analisar a complexidade temporal.

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

ACRESCENTA utilizador1 0000111122223333 2301

ACRESCENTA utilizador1 0000111122223333 2302

ACRESCENTA utilizador1 0000111122223334 2401

ACRESCENTA utilizador2 0000111122223336 2402

ACRESCENTA utilizador2 0000111122223337 2203

CONSULTA utilizador1

CONSULTA utilizador3

LISTAGEM

APAGA

FIM

Saída

NOVO UTILIZADOR CRIADO

CARTAO ATUALIZADO

NOVO CARTAO INSERIDO

NOVO UTILIZADOR CRIADO

NOVO CARTAO INSERIDO

0000111122223333 2302

0000111122223334 2401

FIM

NAO ENCONTRADO

utilizador1 0000111122223333 2302 0000111122223334 2401

utilizador2 0000111122223336 2402 0000111122223337 2203

FIM

LISTAGEM APAGADA

Compreende a elaboração do Relatório 3.3 (formulário fornecido) e vai incidir sobre a análise de complexidade temporal do programa desenvolvido.

O relatório a realizar com base no formulário disponibilizado deve ter em conta:

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

Relatório (a submeter no InforEstudante)

Sub-projeto 3.4: Análise Comparativa das Estruturas de Dados usadas nos subprojetos

Compreende a elaboração do Relatório 3.4 (template fornecido) e vai incidir sobre os seguintes pontos:

1) Análise geral comparativa das várias estruturas de dados usadas (PROS/CONS).
2) Para cada um dos subprojectos descrição da tomada de decisão sobre estruturas de dados para cada um dos subprojetos

Relatório (a submeter no InforEstudante)