



UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA

**Departamento de Engenharia
Informática**

Projeto #4 Algoritmos e Estruturas de Dados

2022-2023 – 2º Semestre

VERSÃO 1.1

Submissão relatório (InforEstudante) e Mooshak:
21 de abril 23:59

Anotações: este projeto foi preparado para ser resolvido parcialmente no espaço das quatro sessões práticas. Será disponibilizado um formulário para o relatório a preencher com a execução 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, sem referenciar as fontes, 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 as 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.

Tarefas

1. Implementação de soluções para os problemas propostos, utilizando uma Árvore Binária de Pesquisa não balanceada
2. Implementação de soluções para os problemas propostos, utilizando a estrutura de dados mais adequada
3. Elaboração de um relatório com a análise das soluções implementadas

Tema

Com a crescente adoção dos programas de conversação assistidos por inteligência artificial, os alunos do DEI decidiram fazer a sua própria solução, o **DEIgpt**. Dentro dos vários sistemas desenvolver encontram-se:

1. Armazenamento de conversas com o DEIgpt
2. Gestão de utilizadores da plataforma
3. Análise de métricas de temas mais recorrentes

1. Sistema de conversas com o DElgpt (S1)

Este sistema é responsável por armazenar os vários *chats*/conversas que os utilizadores iniciam. Para isso deverá permitir que a plataforma insira novas *prompts*¹, agrupadas em conversas; obtenha todas as *prompts* de uma determinada conversa; e apague uma conversa.

Para isso, por cada conversa é necessário registar um identificador único, o nome do utilizador e as várias *prompts*.

Como os utilizadores tendem em usar novas conversas, em vez de usar as já existentes, o sistema faz consideravelmente mais inserções de conversas (criar uma *prompt* numa nova conversa) do que consultas (ex: criar uma *prompt* numa conversa já existe).

O programa a implementar deverá processar um conjunto de comandos recebidos pela consola, e escrever para a mesma o respetivo resultado.

As operações disponíveis são:

- **NEW_PROMPT** *chatId username prompt* – Adiciona uma nova *prompt* na conversa com o *chatId* correspondente. Se a conversa ainda não existir, cria uma. Imprime:
 - Para uma nova conversa: CHAT *chatId* CRIADO
 - Para uma conversa existente: CHAT *chatId* ATUALIZADO
- **GET_CHAT** *chatId* – Imprime o utilizador associado, e lista todas as *prompts* para respetiva conversa. Imprime:
 - Para uma conversa existente:
 - Uma linha com o *username* da conversa
 - Uma linha com cada *prompt*
 - Uma linha com a palavra **FIM**
 - Para uma conversa não existente: CHAT *chatId* NAO ENCONTRADO
- **DELETE_CHAT** *chatId* – Elimina a conversa com o *chatId* correspondente. Imprime:
 - Para uma conversa existente: CHAT *chatId* APAGADO
 - Para uma conversa não existente: CHAT *chatId* NAO ENCONTRADO
- **FIM** – termina a sequência de comandos

Pressupostos:

- **chatId** é um inteiro
- **username** é uma string alfanumérica, não contém espaços
- **prompt** é uma string com um pequeno texto, com no máximo 100 caracteres
- Cada conversa tem apenas 1 username associado

¹ Terminologia usada na indústria para um pedido feito através de texto pelo utilizador

Exemplo:

Input

```
NEW_PROMPT 1 john_doe Como ordenar um array em Python?
NEW_PROMPT 2 jane_doe Como devo entregar a minha dissertacao de mestrado?
NEW_PROMPT 2 jane_doe Qual o horario de abertura da secretaria do DEI?
GET_CHAT 2
NEW_PROMPT 1 john_doe Como resolver o erro "SyntaxError: invalid syntax"?
DELETE_CHAT 1
NEW_PROMPT 1 john_doe2 Quando e a proxima entrega de AED?
DELETE_CHAT 3
FIM
```

Output

```
CHAT 1 CRIADO
CHAT 2 CRIADO
CHAT 2 ATUALIZADO
jane_doe
Como devo entregar a minha dissertacao de mestrado?
Qual o horario de abertura da secretaria do DEI?
FIM
CHAT 1 ATUALIZADO
CHAT 1 APAGADO
CHAT 1 CRIADO
CHAT 3 NAO ENCONTRADO
```

2. Sistema de gestão de utilizadores da plataforma (S2)

Este sistema é responsável por armazenar os utilizadores da plataforma e se inscrevem o plano *premium* ou não. Desta forma será possível a plataforma registar novos utilizadores e determinar se deve gerar respostas com maior ou menor velocidade.

Para isso, o sistema deverá permitir que a plataforma registre novos utilizadores - com nome de utilizador e plano - ou os apague, e obtenha ou atualize os seus planos.

Como este é um sistema ainda interno existem poucas inscrições de novos utilizadores. No entanto, a comunidade do DEI utiliza intensamente o sistema e, como este verifica qual o plano que os utilizadores inscrevem a cada *prompt*, tem muito mais consultas destes dados do que inserções.

As operações disponíveis são:

- **NEW_USER** *username plano* – Adiciona um novo utilizador, e o plano. Imprime:
 - Para um novo utilizador: USER *username* CRIADO
 - Para um utilizador existente - não atualiza o plano: USER *username* JA EXISTE
- **UPDATE_USER** *username plano* – Atualiza o plano associado ao utilizador correspondente. Imprime:
 - Para um utilizador existente: USER *username* ATUALIZADO
 - Para um utilizador não existente: USER NAO ENCONTRADO
- **GET_TYPE** *username* – Imprime o plano associado ao utilizador. Imprime:
 - Para um utilizador existente: Uma linha com o *plano*
 - Para um utilizador não existente: USER NAO ENCONTRADO
- **DELETE_USER** *username* – Remove o registo para o utilizador correspondente. Imprime:
 - Para um utilizador existente: USER *username* APAGADO
 - Para um utilizador não existente: USER NAO ENCONTRADO
- **FIM** – termina a sequência de comandos

Pressupostos:

- **username** é uma string alfanumérica, não contém espaços
- **plano** é uma string com o valor “FREE”, “BASIC” ou “PREMIUM”

Exemplo:

Input

```
NEW_USER john_doe FREE
NEW_USER jane_doe PREMIUM
NEW_USER jane_doe PREMIUM
UPDATE_USER john_doe PREMIUM
UPDATE_USER john_doe2 PREMIUM
GET_TYPE john_doe
GET_TYPE user123
DELETE_USER john_doe
FIM
```

Output

```
USER john_doe CRIADO
USER jane_doe CRIADO
USER jane_doe JA EXISTE
USER john_doe ATUALIZADO
USER NAO ENCONTRADO
PREMIUM
USER NAO ENCONTRADO
USER john_doe APAGADO
```

3. Sistema de análise de métricas de temas mais recorrentes (S3)

Este sistema é responsável por registrar quantas vezes um determinado tópico foi abordado nas *prompts*. Assim será possível aos alunos que desenvolveram o DEIgpt monitorizar quais os tópicos mais abordados e tentarem melhorar a experiência de utilização com base nesses dados.

Para isso, por cada assunto abordado o sistema deverá registar uma entrada com o seu nome e número de vezes que foi abordado.

Durante vários períodos, os tópicos abordados têm tendência a convergir entre os vários utilizadores. Por isso, a sua contabilização pode ser bastante assimétrica – 90% dos incrementos são feitos a 5% dos tópicos.

As operações disponíveis são:

- **ADD_SUBJECT** *tópico* – Incrementa a contabilização de um tópico. Se o tópico não existir, cria um. Imprime:
 - REGISTADO
- **GET_SUBJECT_COUNT** *tópico* – Imprime a contabilização atual para o tópico correspondente. Imprime:
 - Para um tópico existente: *tópico* N, onde N é um inteiro com a contabilização atual
 - Para um tópico não existente: SUBJECT NAO ENCONTRADO
- **LIST_ALL** – Imprime todos os tópicos registados por ordem alfabética, e as suas contabilizações atuais. Imprime:
 - Uma linha para cada tópico: *tópico* N
 - Uma linha com a palavra **FIM**
- **FIM** – termina a sequência de comandos

Pressupostos:

- **tópico** é uma string alfanumérica, não contém espaços

Exemplo:

Input

```
ADD_SUBJECT Mestrado  
ADD_SUBJECT Mestrado  
ADD_SUBJECT Python  
ADD_SUBJECT Mestrado  
GET_SUBJECT_COUNT Mestrado  
GET_SUBJECT_COUNT Python  
LIST_ALL  
FIM
```

Output

```
REGISTADO  
REGISTADO  
REGISTADO  
REGISTADO  
Mestrado 3  
Python 1  
Mestrado 3  
Python 1  
FIM
```

Relatório

O relatório deve ter como base o template que será disponibilizado na plataforma UCStudent, e deverá ser entregue em formato PDF no InforEstudante até dia 21 de abril às 23:59.

O relatório consiste em:

- Análise dos tempos de execução de cada solução;
- Análise do número de operações executadas para cada solução;
- Tabelas concisas com as medições efetuadas;
- Gráficos com as medições, incluindo o resultado das regressões;
- Reflexão crítica sobre os resultados e possíveis outliers;
- Análise de complexidade com base nos resultados empíricos obtidos;
- Reflexão sobre as estruturas de dados usadas.