

USP – ICMC – SSC

SSC0603 – Estrutura de Dados I (ED1) – 2023/2
TRABALHO TP02 – Big Brother – Revisado R01

Árvores Binárias – Big Brother

Versão 0.1r01 - VER sempre o PDF com descrição completa!

ATENÇÃO: DATA LIMITE DE UPLOAD 08/12/23 21:00:00

(Fecha a submissão ANTES da data/hora marcada no RunCodes, usaremos ESTA data!)

> Usando os TADs Lista Dinâmica Encadeada Dupla - LDED (Lista Dupla com Ordenação) e ArvBin ABO (Árvore Binária Ordenada NÃO BALANCEADA)

Contexto/Tema do Trabalho:

- Em um tempo não muito distante um governo autoritário inspirado no Big Brother de George Orwell, adotou um controle universal identificando as pessoas por um TAG-Identificador único, denominado de CPF (“Controle de Pessoas Fichadas”). Esse TAG permite associar e monitorar informações das pessoas como Nome, Data de Nascimento, Endereço, Nro. do Telefone (celular) e sua GeoLocalização, Acesso à Internet (Login usando o CPF), Registro de horário de trabalho e de estudos (Carteira de Identificação com RFID-CPF para acesso aos locais de estudo e trabalho), Registro de condutor de veículos (associado a placa do mesmo) ou de uso de carros de aplicativo, Controle de seus hábitos de consumo (alimentação, compras, etc). Em resumo, o registro do TAG-Identificador único – CPF, permite monitorar TUDO sobre a vida das pessoas, e assim, **saber se ela tem algum tipo de “dívida”** para com o Estado Central Autoritário (ECA), podendo ser devido a uma dívida monetária ou a algum tipo de **comportamento inadequado, ilícito ou multa devida**: fez o que não devia, foi onde não devia ir, agiu de modo inadequado de acordo com as normas ECA.
- Qualquer semelhança com o mundo atual é “pura coincidência”. Este trabalho é uma “obra ficcional” assim com a obra de George Orwell...

Descrição do Trabalho:

> Criar um programa que armazena de **modo ordenado “TAGs” de identificação e monitoramento de pessoas**. Os TAGs são alfanuméricos, contendo **4 letras** (‘A’ a ‘Z’) seguidas de **4 dígitos** (‘0’ a ‘9’), por exemplo: **ABCD0123** e um controle numérico indicando se a pessoa é ou não “devedora” e “procurada” pelas forças de ordem do Governo Big-Brother (valor **0 = livre**; **1 = procurado**). Esta base de dados (estrutura de dados) unifica as informações de busca e apreensão de indivíduos procurados: TAG-ID + Status-Procurado.


> Atenção: as TAGs têm um tamanho fixo de **8 caracteres, 4 letras maiúsculas e 4 dígitos**. Atenção: cada TAG tem associado um “Status”, valor numérico 0 (ok) ou 1 (procurado). Atenção: serão criadas 2 estruturas de dados na memória, uma **lista encadeada dupla** e **uma árvore binária ordenada** (não balanceada), para armazenar os mesmos dados.

> Entradas:

1. **Arquivo**: ler um arquivo com nome “**arq.txt**” contendo os dados a serem inseridos nas estruturas de dados (nodo: TAG e Status), conforme formato descrito a seguir.
O arquivo termina com uma linha com “x” (minúscula) no lugar do TAG.
2. Ler a **entrada do teclado** onde será fornecido um **comando** (numérico) e um **TAG**, realizando a operação descrita pelo tipo de comando conforme descrito a seguir.

> Saídas:  **antes, já passar todos os dados para as estruturas**

Exibir na tela os resultados do comando, conforme cada comando, e conforme especificado a seguir. Por exemplo:

Comando 1 – Consulta TAG: Exibe se achou a TAG, o status da TAG,
 o nro de nodos consultados na lista encadeada,
e depois o nro. de nodos consultados na árvore binária.

Exemplo: Entrada do teclado => **1 ABCD0123**

Saída na tela => S:0 NL:10 NA:2

Significa: S:0 ou S:1 significa que achou (0 tem status OK)

NL nodos visitados Lista (10 nodos)

NA nodos visitados Arvore (2 nodos)

Outra possível saída na tela => S:-1 NL:10 NA:2

Significa: S:-1 Significa que não achou (-1)

NL nodos visitados Lista (10 nodos, e não achou)

NA nodos visitados Arvore (2 nodos, e não achou)

>> FORMATO DO ARQUIVO DE ENTRADA (Arquivo “arq.txt”):

O arquivo texto é composto sempre de **pares de linhas** de entrada com uma **string** (4 letras maiúsculas e 4 dígitos) que formam o conteúdo da **TAG**, seguido de uma linha com o **Status** numérico (**0** ou **1**).
O TAG e Status são o conteúdo a ser armazenado nos nodos da lista e da árvore.
O arquivo termina com um TAG inválido, ou seja, com “x” (minúsculo).
Exemplo: > Cria uma lista ordenada/árvore ordenada com 4 nodos.

```
BBBB1234
1
XPT09876
1
ABCD0123
0
BIGB0666
0
x
```

>> COMANDOS DE ENTRADA (Teclado):

O programa aceita **apenas 1 comando por execução**, ou seja, é lido o comando do teclado, executado, e a **execução do programa já é encerrada** (sai do programa).

Comando 0 (zero): Lê apenas o valor 0, sem parâmetros adicionais.

Exemplo – Entrada do teclado (número): 0

Execução: Exibe o total de nodos que foram inseridos na Lista Encadeada e na Árvore Binária

Saída na tela => NL:4 NA:4 (Para o exemplo do arquivo acima fornecido de 4 nodos)

já tem o numero de
nodos da arvore e
de elementos da
lista



Comando 1 (um): Lê o valor 1, seguido do parâmetro adicional que é a identificação do TAG
Este comando procura e exibe o status do TAG informado, e o nro. de nodos visitados.

Exemplo 1 – Entrada do Teclado: 1 ABCD0123

Considerando a árvore criada com o arquivo exemplo mostrado acima de 4 nodos

Saída na tela => S:0 NL:1 NA:2

Significa: S:0 significa que achou (valor 0 tem status OK)

NL **nodos visitados** Lista Ordenada (1 nodo)

NA **nodos visitados** Arvore Ordenada (2 nodos)

Exemplo 2 – Entrada do Teclado: 1 ZZZZ0000

Considerando a árvore criada com o arquivo exemplo mostrado acima de 4 nodos

Saída na tela => S:-1 NL:4 NA:2

Significa: S:-1 Significa que não achou (-1)

NL nodos visitados Lista (4 nodos, e não achou)

NA nodos visitados Arvore (2 nodos, e não achou)

então, ao inves de colocar
como modo de parada o
NULL, colocar se o nodo é
maior --> para e retornar -1
por exemplo

Atenção:

Como a lista encadeada é ordenada, a busca para quando já não for mais possível achar o nodo, ou seja, **se já passou um nodo menor que o procurado, e os outros nodos são maiores na ordem, não tendo mais como achar o nodo procurado, pois os nodos estão ordenados!** Para a busca.

Na contagem, inclui o nodo que é imediatamente maior, pois precisa consultar/visitar este nodo para concluir que não existe o nodo procurado.

A saída na tela deve ser EXATAMENTE como apresentado acima, por exemplo,

S:0 NL:1 NA:2

“S” maiúsculo, dois pontos, 0 (zero, sem espaço antes do zero e após o ‘:’), espaço (um só ‘ ‘),

“NL” maiúsculo, dois pontos, 1 (um sem espaço antes e após o ‘:’), espaço (um só espaço),

“NA” maiúsculo, dois pontos, 2 (dois, sem espaço antes e após o ‘:’) e “nova linha” (‘\n’).

Comando 2 (dois): Exibe a **lista encadeada** no formato descrito a seguir

Entrada do teclado (número): 2

Execução: Exibe apenas os nodos da Lista Encadeada Ordenada (em ordem)

Saída na tela: Para o exemplo do arquivo acima fornecido de 4 nodos (serve para “debug”)

```
ABCD0123 0
BBBB1234 1
BIGB0666 0
XPT09876 1
```



Comando 3 (três): Exibe a **árvore binária** ordenada no formato descrito a seguir **(pré-ordem)**

Entrada do teclado (número): 3

Execução: Exibe apenas os nodos da Árvore Binária Ordenada em modo “pre-fixado” (pré-ordem)

Saída na tela: Para o exemplo do arquivo acima fornecido de 4 nodos (serve para “debug”)

```
BBBB1234 1
ABCD0123 0
XPT09876 1
BIGB0666 0
```