

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Estrutura de Dados I

Trabalho 2

Relatório: Amigos de graduação indo ao cinema

Alunos

Gustavo Okuvo - 10818911

Kalwin Vianna Alves - 10786062

Stefane Tame Monteiro Oliveira - 10829970

Prof. Dr. Thiago A. S. Pardo

São Carlos – 2019/2

1. Introdução

O trabalho consiste na criação de um sistema de cadastro de alunos. Esse sistema será trabalhado em uma árvore binária de busca. Cada nó dessa árvore contém o número usp, que será a chave indicativa; o nome do aluno; uma lista de filmes prediletos, onde haverá no mínimo um filme contido; e dois indicadores de direita e de esquerda.

A intenção dessa árvore é trabalhar com a relação de filmes entre os alunos, ou seja, os alunos poderão encontrar outros com filmes em comum, com nenhum filme em comum, ver todos os filmes prediletos dos alunos, bem como outras operações disponíveis.

Este relatório contém: uma breve explicação de como foi desenvolvido o algoritmo e o funcionamento de cada função exigida; um relato das estruturas de dados utilizadas e suas justificativas; e uma seção detalhando toda a compilação do programa.

2. Desenvolvimento

A maneira como foi implementado o trabalho foi utilizando TAD (Tipo Abstrato de Dados), que é um método que abstrai os dados e opera-os de modo eficaz, sem que os usuários possam interferir. Sendo assim, foram criados três arquivos para essa abstração: `main.c`, `trabalho.c` e `trabalho.h`.

No arquivo `main.c`, há toda a interface do sistema criado. A interface foi criada de modo a ficar claro para o usuário todos os passos e todas as operações disponíveis. Há, também, o menu, que é a função que verifica o que o usuário deseja fazer na vez e recolhe todos os dados necessários, caso necessite; dentro de cada caso do menu, há verificações de erros encontrados ao longo da função utilizada e imprime para o leitor o sucesso ou o fracasso do caso que o usuário optou.

No arquivo `trabalho.c`, há todas as funções, em que algumas são utilizadas na `main`, isto é, que são utilizadas pelo usuário. Nem todas são utilizadas, pois são funções auxiliares das funções principais. A seguir, todas as funções utilizadas no `trabalho`:

a) Função de alocação para a árvore.

b) Função de criar um cadastro: se a árvore estiver vazia insere na raiz; caso contrário, percorre de forma inteligente e recursiva para a inserção, ou seja, se a chave for menor que a chave do nó atual, percorre pela sub-árvore esquerda e se for maior, percorre pela sub-árvore direita. Quando encontrado o local certo, insere o número usp, o nome do aluno, aloca uma lista de filmes para este nó (utilizando de uma função que aloca uma lista) e insere todos os filmes nesta lista, sem repetição (utilizando de uma função que insere na lista de forma ordenada). Não é cadastrado o aluno que já tiver sua chave inserida na árvore. Usa de uma função auxiliar que recebe a árvore para inicializar a função principal recursiva que recebe o nó.

c) Função de listar alunos cadastrados no sistema: imprime todos os nomes dos alunos e seus números usp em ordem, ou seja, percorre a sub-árvore esquerda, raiz e por último a sub-árvore direita. Usa de uma função auxiliar que recebe a árvore para inicializar a função principal recursiva que recebe o nó.

d) Função de verificar um usuário específico no sistema: percorre a árvore toda de forma inteligente e recursiva, ou seja, se a chave for menor que a chave do nó atual, percorre pela sub-árvore esquerda e se for maior, percorre pela sub-árvore direita. Se encontrar, retorna 1; caso contrário, retorna 0. Usa de uma função auxiliar que recebe a árvore para inicializar a função principal recursiva que recebe o nó.

e) Função de listar todos os filmes do sistema: cria uma lista que irá conter os filmes do sistema e inicializa uma função que percorre todos os nós da árvore, inserindo na lista criada; a função de inserir em lista não insere filmes repetidos. Após isso, retorna para a função inicial e imprime todos os filmes da lista.

f) Função de verificar um filme específico no sistema: percorre todos os nós da árvore e verifica se o filme escolhido está no nó atual, usando uma função de lista que retorna 0, se o filme não estiver na lista, e retorna 1, caso contrário. Usa de uma função auxiliar que recebe a árvore para inicializar a função principal recursiva que recebe o nó.

g) Função de encontrar o aluno com mais filmes em comum com outro: primeiramente, inicializa uma função que retorna a quantidade de nós na árvore, ou seja, a quantidade de alunos cadastrados para alocação de vetores auxiliares e já

verifica se há 0 ou 1 cadastros no sistema para retornar erro. Se não retornar erro aqui, continua alocando os vetores auxiliares. Encontra o nó do aluno que quer achar outro aluno com filmes em comum, utilizando da função que retorna o nó, dado um número usp. Inicializa a função que acha todas as quantidades de filmes em comum, onde percorre nó por nó; a cada nó, é inicializada outra função que percorre a lista de filmes do nó do aluno principal e verifica se cada um está na lista do nó atual, utilizando da função que verifica se um filme está na lista, e assim inicia o contador. Após percorrer toda a árvore, retorna para função principal e acha o maior valor no vetor de quantidades; se o maior valor for 0, retorna erro; caso contrário, prossegue achando o nome do aluno encontrado, a partir da função que acha o nó usando um número usp.

h) Função de encontrar o aluno com menos filmes em comum com outro: esta função desenvolve da mesma maneira que a anterior. No entanto, difere-se ao ter que encontrar o menor valor no vetor de quantidades; se o menor valor for igual a quantidade de filmes do aluno principal, retorna erro; caso contrário, prossegue achando o nome do aluno encontrado, a partir da função que acha o nó usando um número usp.

i) Função de gerar um arquivo com as informações do sistema: cria um arquivo e inicializa outra função que percorre a árvore, nó por nó, em ordem, ou seja, percorre a sub-árvore esquerda, raiz e por último a sub-árvore direita. A cada nó, insere o nome do aluno, seu número usp e seus filmes prediletos.

j) Função de imprimir os dados da árvore do sistema: imprime o número de nós da árvore, a altura da árvore e a maior diferença de alturas de todas sub-árvores da árvore principal, utilizando de funções respectivas que calculam o que é pedido.

k) Função de remover um aluno cadastrado: primeiramente, percorre a árvore toda de forma inteligente e recursiva, ou seja, se a chave for menor que a chave do nó atual, percorre pela sub-árvore esquerda e se for menor, percorre pela sub-árvore direita, até encontrar a posição da chave a ser removida. Quando encontra, faz 3 verificações do nó: se não tem filhos, se tem um filho só ou se tem dois filhos; e é feita a operação conforme sua característica. Usa de uma função

auxiliar que recebe a árvore para inicializar a função principal recursiva que recebe o nó.

l) Função de inserir um filme em algum aluno cadastrado: caso o aluno, obtenha um filme predileto, ele pode inserir usando essa função, que inicializa outra de inserção na lista de filmes associada.

m) Função de remover um filme em algum aluno cadastrado: caso o aluno queira retirar o filme, desde que não zere sua lista de filmes, ele pode remover algum filme de sua lista, utilizando de outra função que remove da lista de filmes associada.

n) Função de desalocação da árvore: não esquecendo de desalocar cada lista associada a cada nó da árvore. Usa de uma função auxiliar que recebe a árvore para inicializar a função principal recursiva que recebe o nó.

E, por fim, no arquivo trabalho.h, há todas as funções disponíveis para o usuário utilizar. Também há os registros utilizados, caso seja necessário para o usuário.

3. Estrutura de dados

As estruturas de dados utilizadas foram árvore binária de busca e lista ordenada, ambas implementadas de forma encadeada e dinâmica; tudo conforme as exigências do trabalho.

A utilidade de ser árvore binária de busca para implementar o sistema de cadastros, é pela complexidade de suas operações, que é muito mais otimizada que uma árvore comum.

Árvore binária de busca		
Tipo	Árvore	
Ano	1960	
Inventado por	P.F. Windley, A.D. Booth, A.J.T. Colin, e T.N. Hibbard	
Complexidade de Tempo em Notação big O		
Algoritmo	Caso Médio	Pior Caso
Espaço	$O(n)$	$O(n)$
Busca	$O(\log n)$	$O(n)$
Inserção	$O(\log n)$	$O(n)$
Remoção	$O(\log n)$	$O(n)$

Figura 1: Complexidade da ABB

A utilidade de ser lista ordenada é pela organização dos dados, ou seja, pela ordenação alfabética dos filmes.

A vantagem de ser implementado de forma encadeada e dinâmica é pela fácil manipulação de dados, isto é, manipulação de indicadores (ponteiros).

4. Compilação

O trabalho foi compilado usando o terminal do sistema operacional linux, onde rodou perfeitamente o algoritmo do trabalho. A versão do terminal é a seguinte:

```
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.6 LTS"
NAME="Ubuntu"
VERSION="16.04.6 LTS (Xenial Xerus)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 16.04.6 LTS"
VERSION_ID="16.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
VERSION_CODENAME=xenial
UBUNTU_CODENAME=xenial
```

Figura 2: Versão do terminal, utilizado para compilar o programa

As bibliotecas utilizadas foram `stdio.h`, `stdlib.h`, `string.h` e o próprio trabalho `criado trabalho.h`.

O processo de compilação é o seguinte:

```
stefane@stefanetame:~/ED1/TRABALHO 2$ gcc main.c -c
stefane@stefanetame:~/ED1/TRABALHO 2$ gcc trabalho.c -c
stefane@stefanetame:~/ED1/TRABALHO 2$ gcc main.o trabalho.o -o sistema.exe
stefane@stefanetame:~/ED1/TRABALHO 2$ ./sistema.exe
Operacoes disponiveis para o sistema de cadastros:
1) Cadastrar aluno
2) Listar alunos e seus numero usp
3) Verificar se um aluno X esta cadastrado
4) Listar todos filmes prediletos dos alunos
5) Verificar se um filme X esta no cadastro de algum aluno
6) Identificar algum aluno que tem mais filmes em comum com o aluno X
7) Identificar algum aluno que tem menos filmes em comum com o aluno X
8) Produzir um arquivo com as informacoes cadastradas no sistema
9) Exibir (arvore de cadastros): numero de nos; a altura; maior diferenca de alt
uras
10) Retirar algum aluno cadastrado
11) Inserir filme predileto do aluno X
12) Remover filme predileto do aluno X
13) Sair
0 que deseja fazer? (Escolha uma das opcoes acima entre 1 e 13)
```

Figura 3: Compilação do programa

Primeiramente, compila-se a main.c e o trabalho.c e é gerado um objeto de cada um. Após isso, gera um executável com os objetos criados. E, por fim, executa o programa, no qual imprime a interface de operações.

5. Referências

[1] https://pt.wikipedia.org/wiki/%C3%81rvore_bin%C3%A1ria_de_busca