

# Árvores e Buscas

13 de julho de 2021

TRABALHO - 2

## 1 Apresentação do Problema

Dois amigos, João e Vinicius, conversam sobre estruturas de árvores binárias e suas apresentações dentro do ambiente da empresa em que trabalham - as formas como elas podem ser mostradas para o usuário ou salvas em arquivo para serem reaproveitadas por outros processos ou programas.

O primeiro amigo, João, acredita que a impressão dos elementos em ordem crescente é a melhor forma de apresentar os dados, porque visualmente é mais agradável e também porque o próximo programador poderá escolher com qual estrutura irá armanar os dados.

O segundo amigo, Vinicius, acredita que a árvore deve ser mostrada por níveis e que antes de mostrar um segundo nível, é importante mostrar todos os elementos que estão no nível anterior, pois assim temos uma noção do quanto os valores crescem na nossa árvore.

João e Vinicius são responsáveis pelo cadastro de duas áreas diferentes, Pessoas e Empresas, respectivamente. Como ambos estão implementando árvores binárias balanceadas, eles decidiram utilizar a mesma estrutura para pouparem tempo na implementação. A partir da estrutura implementada, cada um implementou o seu próprio método para impressão dos dados coletados.

**Problema:** Sua missão é implementar uma árvore binária balanceada que consiga armazenar ambas as estruturas necessárias, mas não as duas ao mesmo tempo. E também implementar as impressões dos dados segundo

as preferências de João (imprimir todos os elementos em ordem crescente) e Vinicius (imprimir todos os elementos de cada nível da árvore antes de imprimir o elemento do próximo nível).

**Entrada:** O responsável pela coleta de dados, Lauro, precisou levar os questionários preenchidos para casa e seu filho, quando cansou de brincar com os discos, resolveu brincar de ler as fichas do papai. Como resultado, Lauro misturou os dados das empresas e das pessoas no mesmo arquivo de entrada. Assim, a entrada identifica a quantidade de itens totais (Pessoa ou Empresa) e depois identifica cada um dos itens. Portanto a entrada é constituída de um inteiro  $n$ , seguido de  $n$  itens que pode ser do tipo Pessoa ou Empresa. O tipo Pessoa contém os seguinte dados, nesta ordem: Identificador (char), Nome completo (string com até 50 letras), idade (int) e altura (float).

*P*  
*Boris*  
7  
0.8

O tipo Empresa contém os seguinte dados, nesta ordem: Identificador (char), Cnpj (apenas números), Nome (string com até 50 caracteres) e balanço do último ano (float).

*E*  
1021321310123  
*Empresa do papai*  
5000000.99

**Entrada:**

$n$   
 $T_1$   
 $T_2$   
 $\dots$   
 $T_n$

Onde cada  $T_i$ ,  $1 \leq i \leq n$ , é uma descrição de Pessoa ou Empresa.

Exemplo de entrada:

5  
*P*  
*Boris*  
 7  
 0.8  
*E*  
 1021321310123  
*Empresa do papai*  
 5000000.99  
*P*  
*Olivia*  
 5  
 0.7  
*P*  
*Momo*  
 3  
 1.1  
*E*  
 1021321310122  
*Empresa da mamae*  
 5000001.99  
*E*  
 1021321310124  
*Empresa da titia*  
 10001.99

No exemplo acima, o a entrada possui 6 itens, com 3 pessoas e 3 empresas.

**Saída:** A sua saída será constituída pelas pessoas ordenadas pela idade (utilizando a forma de impressão de João) seguidas das empresas ordenadas pelo balanço do último ano (utilizando a forma de impressão de Vinicius).

*P*  
*Momo*  
3  
1.1  
*P*  
*Olivia*  
5  
0.7  
*P*  
*Boris*  
7  
0.8  
*E*  
1021321310123  
*Empresa do papai*  
5000000.99  
*E*  
1021321310124  
*Empresa da titia*  
10001.99  
*E*  
1021321310122  
*Empresa da mamac*  
5000001.99

É uma sequência que apresentamos os dados segundo João e Vinicius querem.

## 2 Avaliação

O seu trabalho será avaliado seguindo os seguinte critérios:

- 1) Os resultados impressos que atendem aos critérios estabelecidos por João.
- 2) Os resultados impressos que atendem aos critérios estabelecidos por Vinicius.
- 3) Seu código deve estar modularizado, com divisões dos arquivos *.h* e *.c* e estrutura encapsulada;

- 4) O arquivo compactado deve conter os arquivos todas as estruturas utilizadas, o arquivo hanoi.c e o arquivo Makefile;
- 5) O arquivo Makefile deve conter ser capaz de compilar o código utilizando o comando “make” (ou seja, utilizando o armento all) e também deve ser capaz de eliminar todos os arquivos que foram criados na compilação utilizando o comando “make clean”;
- 6) O código deve estar comentado (comentários são importante para outros entenderem o seu código);
- 7) O código deve estar modularizado (impacta a reusabilidade);
- 8) Atente-se para os critérios de legibilidade do código, dando nomes legíveis para as funções e variáveis.

### 3 Entrega

**Valor:** 24 pontos.

**Data de Entrega:** Seu trabalho deve ser entregue até às 23:59 do dia 31/07/2021.

**Forma de Entrega:**

- 1) Compacte os arquivos fonte juntamente com o Makefile em um arquivo compactado para o email: joao.thompson@ifes.edu.br;
- 2) utilize o comando: “tar -zcvf nomeSobrenome.tar.gz \*” para compactar todos os arquivos dentro de uma pasta - onde você substituirá nome pelo seu primeiro nome; e Sobrenome para o seu último nome, sem caracteres especiais;
- 3) O arquivo “nomeSobrenome.tar.gz” não deve conter arquivos “.o” ou executaveis;
- 4) Após o envio do trabalho, você receberá a confirmação do recebimento em até 2 dias úteis, caso isso não ocorra, reenvie o trabalho para o mesmo email.
- 5) O assunto do email deve ser “ed:trab2:nomeSobrenome”.

Exemplo:

Para: joao.thompson@ifes.edu.br

De: João da Silva

Assunto: ed:trab2:joaoSilva

Anexo: joaoSilva.tar.gz

### 4 Atenção

O trabalho será compilado e testado utilizando o gcc no sistema operacional Linux. Portanto, por mais que você escreva seu código em outro sistema,

indico que testem em um ambiente Unix.

Trabalhos que não compilam recebem nota ZERO.

Os trabalhos serão verificados automaticamente por uma ferramenta de detecção deplágio. Em caso de detecção de cópia (parcial ou integral), todos os envolvidos recebem nota ZERO. Em outras palavras, tanto os alunos que copiaram quanto o que deixou copiar recebem ZERO. Esta verificação não é realizada apenas com os nomes das variáveis, mas também com os arquivos intermediários gerados na compilação, de forma que apenas mudar uma operação de lugar ou nomes de variáveis não fazem diferença no código.

Trabalhos enviados fora do prazo serão penalizados, corrigidos apenas em caso de prova final - para atingir a nota mínima para atingir a média.

## 5 Dicas

Utilize a estrutura de Pilha para modelar os pinos e a estrutura de Fila para armazenar os movimentos realizados.

Não deixe para fazer o trabalho de última hora, não vai dar tempo.

Existem soluções parecidas na internet, cuidado ao copiar, seu colega vai pesquisar e achar da mesma maneira que você.