

Trabalho 1 CI1057 - Árvores AVL

Eduardo Cunha de Almeida
Paulo Ricardo Lisboa de Almeida

1º Semestre - 2023

1 Descrição

Implemente uma árvore AVL utilizando a linguagem de programação C.

2 Template

Os algoritmos devem ser implementados utilizando como base o template disponibilizado no Moodle.

O arquivo *avl.h* contém os protótipos das funções que obrigatoriamente devem ser implementadas, no arquivo *avl.c*.

Não é permitida a mudança nos protótipos de funções disponibilizados no arquivo *avl.h*. É permitida a criação de funções auxiliares se necessário. Também é permitida a criação de arquivos de header (.h) e de implementação (.c) complementares se necessário.

Você deve alterar a struct definida no arquivo de template para que ela represente corretamente um nodo de AVL. A chave dos nodos deve ser um inteiro.

O programa deve compilar sem erros ou avisos através do comando *make*.

O nome do binário gerado deve ser *trab1_GRR1_GRR2* (e.g., *trab1_1234_5678*). Ajuste isso no *makefile*. Deve ser possível compilar e executar o binário nas máquinas do DInf.

3 Formato de Entrada e Saída

A entrada será feita pela entrada padrão (stdin). A entrada será uma sequência de linhas, onde cada linha representa uma operação. As entradas podem ser as seguintes:

- inserir um valor na árvore: i val
- remover um valor na árvore: r val
- buscar um valor na árvore: b val
- imprimir a árvore em ordem: e
- imprimir a árvore em largura: l

- finalizar e sair: f

Por exemplo, *i 10* significa a inclusão da chave 10 na AVL.

A saída será feita pela saída padrão (stdout). Saídas são geradas apenas quando o usuário solicitar para imprimir a árvore, ou buscar um item. Veja um exemplo de execução do programa a seguir, onde os itens em *itálico* são entradas, e em **negrito** são saídas geradas pelo programa. No exemplo, primeiramente é inserida a árvore representada na Figura 1. Depois, o nodo 443 é removido.

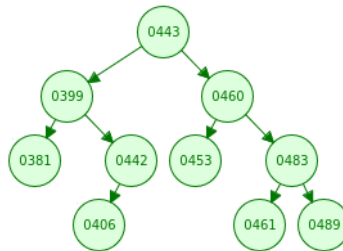


Figura 1 – Representação Gráfica da árvore inserida no programa.

i 443

i 443

Falha ao inserir.

i 399

i 460

i 381

i 442

i 453

i 483

i 406

i 461

i 489

l

[0] 443(0)

[1] 399(1) 460(1)

[2] 381(0) 442(-1) 453(0) 483(0)

[3] 406(0) 461(0) 489(0)

r 443

l

[0] 442(1)

[1] 399(0) 460(1)

[2] 381(0) 406(0) 453(0) 483(0)

[3] 461(0) 489(0)

b 406

Encontrado 406

b 443

Nao encontrado 443.

r 700

Falha ao remover 700.

e

381 399 406 442 453 460 461 483 489

f

A árvore não deve aceitar duplicadas. Ao imprimir em ordem, exiba os valores separados por uma tabulação. Ao exibir em largura, você deve começar cada linha com $[n]$, onde n é o nível atual. Depois, exiba os valores do nível no formato `valor(balanco)`, onde os valores devem estar separados por uma tabulação (veja no exemplo anterior).

A saída do seu programa deve seguir exatamente o modelo disposto neste documento, sob pena de descontos de nota ou até anulação do seu trabalho.

4 Custos das operações

Garanta que os custos dos algoritmos implementados é compatível com a teoria sobre árvores AVL. Por exemplo, inserções, exclusões e buscas devem ter um custo $O(\log_2 n)$, enquanto a impressão da árvore deve ter um custo $O(n)$. Algoritmos ineficientes sofrerão descontos.

5 Exclusão

Para a exclusão, quando necessário transplantar um nodo, utilize o antecessor (o maior filho da subárvore esquerda). Caso você não o faça, seu algoritmo de exclusão poderá ser desconsiderado.

6 Arquivos a serem entregues

Os arquivos devem estar em um diretório de nome `trab1_GRR1_GRR2`. Se, por exemplo, o grr dos alunos é 1234 e 5678, o diretório contendo os arquivos deve se chamar `trab1_1234_5678`. Compacte esse diretório, sendo que a versão compactada vai se chamar `trab1_1234_5678.tar.gz`. Ao descompactar o arquivo `trab1_1234_5678.tar.gz`, deverá ser criado um diretório de nome `trab1_1234_5678` que conterá todos os demais arquivos. O diretório deve conter o seguinte:

- arquivos de código fonte `.c` e `.h` do programa;
- `makefile`;
- `README.txt` com no máximo 6.000 caracteres, contendo a documentação sintetizada do sistema implementado. Qualquer particularidade deve estar descrita neste texto (ex.: como foi implementada a exclusão).

Não inclua quaisquer outros arquivos irrelevantes para o projeto. Por exemplo, não inclua binários compilados, ou arquivos objeto (`.o`). A inclusão de arquivos irrelevantes pode acarretar em descontos de nota.

7 Entrega

O trabalho deve ser entregue via Moodle. A data limite para o envio está estipulada no link de entrega do Moodle. Não serão aceitas entregas em atraso, exceto para os casos explicitamente amparados pelas resoluções da UFPR.

8 Grupos, Pesos e Datas

Grupos: trabalho em duplas.

Valor: 15% da nota do semestre.

Submissão: Via Moodle. Veja a data limite no link de submissão do Moodle.

9 Distribuição da Nota

Alguns descontos padrão, considerando uma nota entre 0 e 100 pontos para o trabalho:

- Plágio: perda total da pontuação para todos os envolvidos. Isso é válido mesmo para casos onde o plágio se refere a apenas um trecho do código;
- Algoritmos implementados em linguagem diferente de C serão desconsiderados;
- Não submissão via Moodle acarreta na perda total dos pontos;
- Falta de algum arquivo requisitado: desconto de 10 a 100 pontos;
- Inclusão de arquivos desnecessários (lixo): desconto de 5 a 20 pontos;
- Erros e avisos de compilação: desconto de 5 a 100 pontos;
- Nomes de arquivo incorretos: 10 pontos por arquivo;
- Arquivo com formato incorreto: 10 a 100 pontos;
- Problemas de programação, como uso de funções descontinuadas, programação insegura, e outros problemas de design: 10 a 100 pontos;
- Corrupção, vazamentos de memória, *user after free*, *double free*, entre outros problemas de gerenciamento de memória: 10 a 100 pontos;
- Formato de saída incorreto: 10 a 100 pontos;
- Algoritmos implementados incompatíveis com uma árvore AVL: 5 a 100 pontos;
- Para cada caso de teste (criado pelo professor, que serão diferentes dos cedidos aos alunos) que o programa falhar, serão descontados de 5 a 10 pontos.

10 Demais Regras

- Dúvidas ou casos não especificados neste documento podem ser discutidos com o professor até a **data de entrega do trabalho**. Não serão aceitas reclamações após a data da entrega.
- Os alunos podem (e devem) procurar o professor para tirar dúvidas quanto ao trabalho.
- O descumprimento das regras dispostas nesse documento podem acarretar na perda parcial ou total da nota do trabalho.