

Algoritmos e Estruturas de Dados II - ACH2024

Exercício Programa II - Operações em Árvores B

Prazo de Entrega: 2/7

1 Descrição

O objetivo deste exercício-programa consiste em implementar operações básicas (inserção, remoção e impressão) em uma estrutura de dados de árvore B com $t = 3$. Nas seções seguintes você entenderá melhor o funcionamento do programa e as estruturas de dados auxiliares que serão necessárias.

2 Desenvolvimento

O programa a ser desenvolvido funcionará como um interpretador de comandos. Cada comando será lido de uma linha de um arquivo de entrada. Os comandos conhecidos pelo interpretador são os seguintes.

- **i <chave>** ¹: insere uma chave na árvore. A chave deve possuir valor numérico inteiro. Esse comando não gera nenhuma saída na tela.

Exemplos:

i 10

i 20

Executando os comandos acima, serão inseridas na árvore B as chaves 10 e 20, respectivamente.

- **r <chave>** : remove da árvore a chave que possui o mesmo valor do parâmetro informado. Esse comando não gera nenhuma saída na tela.

Exemplos:

r 15

r 10

Executando os comandos acima, será removida da árvore B a chave 10 (considerando que a chave 15 não foi inserida previamente na árvore).

- **p**: imprime a árvore atual na saída padrão em ordem, no formato especificado na seção 3 (formato do arquivo de saída).

¹Nos exemplos, os comandos aparecem na cor azul e as respostas esperadas na cor vermelha.

Exemplo:

p
(20)

- **f**: encerra a execução do programa.

Exemplo:

f

3 Execução e formatos de entrada e saída

O programa executável será executado da seguinte forma:

arvoreB.exe <arquivo_de_entrada> <arquivo_de_saida>

- **Arquivo de entrada:** Assume-se que inicialmente a árvore está vazia. Cada linha do arquivo de entrada conterá um comando para manipular essa árvore, inicialmente vazia, conforme formato dos comandos previamente definidos. O último comando será de fim de execução.

Exemplo de arquivo de entrada:

i 30
i 40
i 50
i 60
r 20
r 40
r 30
p
i 20
i 40
i 15
r 15
p
f

- **Arquivo de saída:** Cada comando de impressão **p** do arquivo de entrada deve gerar, no arquivo de saída, uma linha contendo a árvore B, percorrida em ordem, resultante das operações realizadas até aquele momento em formato parentisado. Neste formato, cada nó deve ser representado por um abre e fecha parêntesis, tendo todas as chaves de um mesmo nó separadas por um espaço simples e sub-árvores representadas recursivamente em ordem.

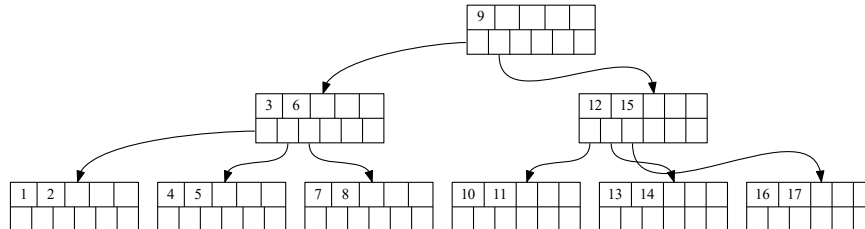
Exemplos:

- nós folhas:
(10 20 30)

– representação parentisada da árvore da Fig 1 percorrida em ordem:

((((1 2) 3 (4 5) 6 (7 8)) 9 ((10 11) 12 (13 14) 15 (16 17))))

Figura 1: Exemplo de árvore resultante com $t=3$.



4 Entrega do trabalho

O trabalho será individual. Deverá ser entregue um único módulo em C, nomeado na forma: `d-<nomeCompleto>.c`, em que `nomeCompleto` corresponde ao nome completo do autor(a). Exemplo: **FernandoChiuHsieh.c**. Este módulo conterá todas as funções implementadas pelo aluno. Incluir nas linhas iniciais do programa o nome e número USP do autor(a).

O código-fonte deverá ser compilável no gcc ou no mingw. Os trabalhos serão compilados via linha de comando para a correção. Por isso, caso utilize algum Ambiente de Desenvolvimento Integrado (IDE) para desenvolver seu programa, antes de entregá-lo verifique se o seu programa compila sem mensagens de alerta e executa corretamente via linha de comando. É requerido apenas a visualização das mensagens solicitadas nos itens acima. Trabalhos com erros de compilação receberão nota zero. Faça todos os testes necessários para garantir que seu programa esteja livre de erros de compilação.

O trabalho deverá ser postado no tidia. O prazo para entrega é 2/7.

Evidência de plágio entre trabalhos não apenas implicará na nota zero no trabalho, como também sujeitará os alunos envolvidos às medidas disciplinares cabíveis.