

Algoritmos e Estruturas de Dados II - ACH2024

Exercício Programa II - Operações em Árvores B+

Prazo de Entrega: 31/07/2021

1 Descrição

O objetivo deste exercício-programa consiste em implementar operações básicas (inserção, remoção e impressão) em uma estrutura de dados de árvore B+ **em memória principal** com grau mínimo $t = 3$.

O programa a ser desenvolvido funcionará como um interpretador de comandos. Cada comando será lido de uma linha de um arquivo de entrada e, dependendo do comando, atualizar a árvore ou produzir uma linha no arquivo de saída. Os comandos reconhecidos pelo interpretador são os seguintes:

- **i <chave>**¹: insere uma chave na árvore. A chave deve possuir valor numérico inteiro e único na árvore. **Não será necessário verificar unicidade de valores.** Esse comando não gera nenhuma saída.

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. Neste EP só serão removidas chaves que existam na árvore.

Exemplo:

r 10

Executando o comando acima, será removida da árvore B+ a chave 10.

- **p**: imprime a árvore atual no arquivo de saída **em uma única linha** em in-ordem, no formato especificado na seção 2 (formato do arquivo de saída). Se a árvore estiver vazia imprimir "Vazia".

Exemplo:

p

(20)

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

Exemplo:

f

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

2 Execução e formatos de entrada e saída

O programa executável será executado da seguinte forma:

```
arvoreBmais.exe <arquivo_de_entrada> <arquivo_de_saida>
```

Ou seja, diferentemente do EP 1, aqui **os nomes dos arquivos de entrada e de saída devem ser obtidos da linha de comando que chamou o programa executável**. Assuma que os arquivos estarão presentes no diretório a partir do qual o programa será executado.

- **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 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+, **com travessia in-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 pré-ordem.

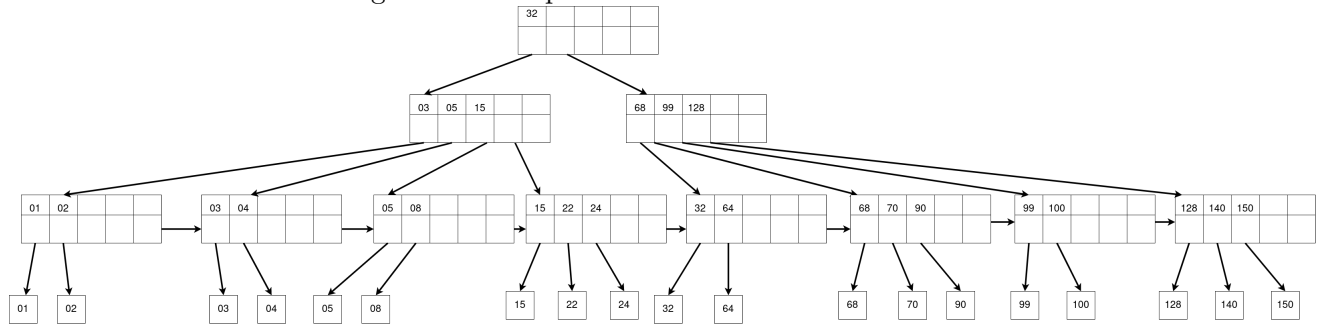
Exemplos:

– nós folhas:
(10 20 30)

– representação parentisada da árvore da Fig 1 percorrida em travessia in-ordem:

```
((((01 02) 03 (03 04) 05 (05 08) 15 (15 22 24)) 32 ((32 64) 68 (68 70 90) 99 (99 100)
128 (128 140 150)))
```

Figura 1: Exemplo de árvore resultante.



3 Entrega do trabalho

O trabalho será individual e deverá ser entregue como um único arquivo .c nomeado na forma:

NomeCompletoTudoJuntoCadaNomeIniciadoComLetraMaiúscula.c

Este arquivo 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. 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.

Evidência de plágio entre trabalhos implicará na nota zero no trabalho.