

2º Trabalho Prático

O programa referente ao 2º trabalho prático da disciplina Organização e Recuperação de Dados deverá ser desenvolvido na linguagem C por grupos de até três pessoas. O código-fonte deverá estar devidamente identificado com os nomes das pessoas do grupo e ser entregue exclusivamente via *Classroom*. Posteriormente, o grupo deverá fazer a apresentação do trabalho em horário previamente agendado.

Especificação

O objetivo deste trabalho é criar um programa que, a partir de um arquivo contendo uma lista de chaves, construa uma **árvore-B**. Além disso, o programa também deverá fornecer funcionalidades para imprimir informações sobre a árvore construída, bem como fazer o percurso na árvore imprimindo as chaves em ordem crescente.

O arquivo de chaves será um arquivo texto contendo apenas chaves numéricas separadas pelo caractere “|”. Cada chave lida (um número inteiro) deverá ser inserida em uma árvore-B de **ordem m** . Para facilitar a experimentação com árvores de diferentes ordens, faça m ser uma constante e use-a ao longo do código sempre que precisar referenciar a ordem da árvore.

A estrutura interna das páginas da árvore de ordem m deve ser similar à vista em aula. **A árvore-B deverá ser criada e mantida em arquivo, logo, a árvore nunca deverá estar completamente carregada na memória.**

O programa deverá oferecer as seguintes funcionalidades:

- Criação da árvore-B a partir do arquivo de chaves;
- Impressão das informações da árvore-B;
- Impressão das chaves em ordem crescente (percurso na árvore-B).

Assim como no 1º trabalho, **as operações serão realizadas sem interação direta com o usuário, utilizando para isso a linha de comando com passagem de parâmetros.**

Criação da árvore-B

A funcionalidade de criação da árvore-B (`-c`) deve ser acessada pela linha de comando, da seguinte forma:

```
$ programa -c nome_arquivo_chaves
```

sendo `programa` o nome do arquivo executável do seu programa e `nome_arquivo_chaves` o nome do arquivo que contém as chaves. Sempre que ativada, essa funcionalidade fará a leitura do arquivo de chaves e a inserção das chaves na árvore-B, que será armazenada no arquivo `btree.dat`. Se o arquivo `btree.dat` já existir, ele deverá ser reescrito. Note que o formato do arquivo de chaves será sempre igual, porém **o número de chaves no arquivo pode variar**. Para simplificar o processamento do arquivo de chaves, considere que ele sempre será fornecido corretamente (i.e., o seu programa não precisa verificar a integridade desse arquivo).

Ao final da criação da árvore-B, o programa deverá apresentar uma mensagem na tela indicando se essa operação foi concluída com sucesso ou não.

Impressão das informações da árvore-B

A funcionalidade de impressão das informações da árvore-B (`-p`) também será acessada via linha de comando, no seguinte formato:

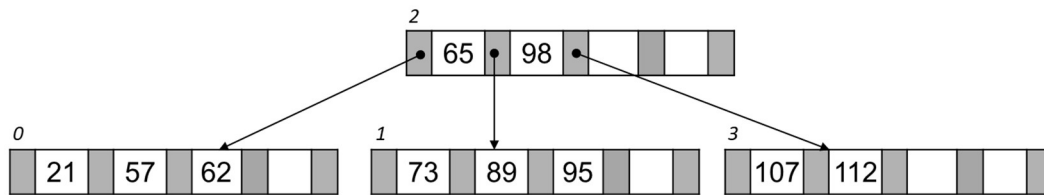
```
$ programa -p
```

sendo `programa` o nome do arquivo executável do seu programa. Note que, para essa execução, o arquivo `btree.dat` deve existir. Caso o arquivo não exista, o programa deve apresentar uma mensagem de erro e terminar.

Sempre que ativada, essa funcionalidade apresentará na tela o conteúdo de todas as páginas da árvore-B armazenada no arquivo `btree.dat`. Para cada página da árvore deverá ser informado: (a) o seu RRN; (b) os valores do vetor de

chaves; e (c) os valores do vetor de filhos. As páginas devem ser apresentadas pela ordem do seu RRN e a página raiz deve ser devidamente identificada.

Como um exemplo, considere a árvore-B construída a partir da inserção das seguintes chaves: 65, 21, 57, 89, 73, 62, 98, 112, 107, 95 (nesta ordem).



Supondo que esta seja a árvore construída, seu programa deverá apresentar as seguintes informações:

```
RRN: 0
Chaves: 21 | 57 | 62
Filhos: -1 | -1 | -1 | -1

RRN: 1
Chaves: 73 | 89 | 95
Filhos: -1 | -1 | -1 | -1

- - - - Pagina Raiz - - - -
RRN: 2
Chaves: 65 | 98
Filhos: 0 | 1 | 3
- - - - -

RRN: 3
Chaves: 107 | 112
Filhos: -1 | -1 | -1
```

Impressão das chaves em ordem crescente (percurso na árvore-B)

A funcionalidade de impressão das chaves em ordem crescente (percurso na árvore-B) (-k) também será acessada via linha de comando, no seguinte formato:

```
$ programa -k
```

sendo `programa` o nome do arquivo executável do seu programa. Note que, para essa execução, o arquivo `btree.dat` deve existir. Caso o arquivo não exista, o programa deve apresentar uma mensagem de erro e terminar.

Sempre que ativada, essa funcionalidade deverá percorrer a árvore-B e apresentar na tela as chaves visitadas. O **percurso deve ser feito de forma que as chaves sejam apresentadas em tela em ordem crescente**. Dica: utilize a lógica do percurso *in-ordem*, o qual, para uma dada chave, visita e imprime sua sub-árvore à esquerda (recursivamente in-ordem) antes de imprimir a própria chave e seguir com a travessia da mesma maneira para a sua sub-árvore à direita.

Supondo a árvore mostrada anteriormente, seu programa deverá apresentar as seguintes informações:

```
21|57|62|65|73|89|95|98|107|112|
```

Qualquer dúvida referente a esta especificação deverá ser sanada com os respectivos professores.

BOM TRABALHO!