

Exercício-Programa 2: Árvores B (Inserção)

Objetivo:

Desenvolver um programa para inserir chaves numéricas em uma árvore B, utilizando as definições e implementações sugeridas em Cormen et al (slides em www.each.usp.br/lauretto/ACH2024_2014/btree.pdf)

A chamada do programa será:

```
btree.exe <arquivo_requisicoes> <arquivo_btree>
```

ou

```
java btree <arquivo_requisicoes> <arquivo_btree>
```

Exemplo:

```
btree.exe req1.txt btree1.txt
```

Entrada:

O arquivo **<arquivo_requisicoes>** será o arquivo de entrada do programa, e conterá a especificação da árvore B juntamente com as chaves a serem inseridas.

A primeira linha conterá a ordem da árvore (denotada por t nos algoritmos apresentados em Cormen et al) e a quantidade n de chaves a serem inseridas. Esses campos serão separados por espaço.

A segunda linha conterá as n chaves a serem inseridas, separadas por espaços. As chaves serão do tipo *float*, com precisão menor ou igual a 6 dígitos.

O exemplo abaixo ilustra um arquivo de requisições para a inserção de 30 chaves em uma árvore B de ordem 3:

```
3 30
29.4 95.2 37.1 52.3 75.5 34.4 15.8 65.1 63 2.6 28.2 24.5 13.3 5.5 65.1 93.6
82.3 68 41 55.5 28.4 33.9 72 62.5 6.9 18.9 22.4 87.6 24.2 74.7
```

Saída:

O arquivo **<arquivo_btree>** conterá a árvore B gerada a partir da inserção das chaves de entrada.

A primeira linha conterá o número de nós da árvore.

Cada uma das demais linhas corresponderá a um nó, e deverá conter as seguintes informações separadas por espaços:

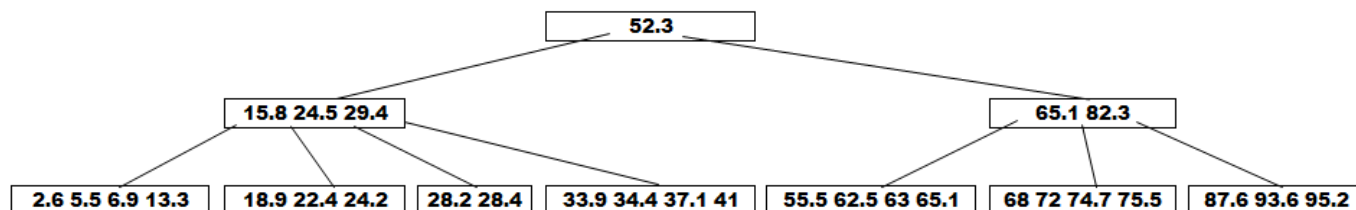
- número de chaves armazenadas no nó;

- valores das chaves do nó.

Importante:

Os nós deverão ser impressos ordenados por profundidade pelos valores de suas chaves. Ou seja, primeiramente será impresso o nó raiz; em seguida, os nós de profundidade 1 (em ordem crescente de chaves); em seguida os de profundidade 2; e assim por diante, até as folhas.

Para o arquivo de exemplos acima, a árvore gerada deverá ser conforme a figura abaixo:



e o arquivo de saída para essa árvore B terá o seguinte conteúdo:

```
10
1 52.3
3 15.8 24.5 29.4
2 65.1 82.3
4 2.6 5.5 6.9 13.3
3 18.9 22.4 24.2
2 28.2 28.4
4 33.9 34.4 37.1 41
4 55.5 62.5 63 65.1
4 68 72 74.7 75.5
3 87.6 93.6 95.2
```

Entrega do trabalho:

Condições da entrega:

- O trabalho é opcional. Detalhes sobre prazos da entrega e bônus a ser concedido estão descritos na página da disciplina.
- O trabalho poderá ser feito em grupos de ATÉ dois alunos, devidamente identificados no nome do arquivo a ser enviado, bem como na primeira linha do código-fonte.
- O programa pode ser implementado em C (compilável com o gcc/mingw) ou em Java (compilável com o jdk).

- Para implementação em C, deverá ser entregue um único módulo contendo a implementação completa do programa, nomeado na forma *d<numerousp1_numerousp2>.c*
Ex: d1234567_7654321.c
- Para implementação em Java, deverá ser entregue um arquivo compactado (formato .zip) nomeado na forma *d<numerousp1_numerousp2>.zip*. Esse arquivo deverá conter o código fonte e o bytecode compilado do programa. Caso o arquivo compactado contenha outros subdiretórios, é obrigatório que a classe principal compilada (de nome *btree*) esteja no diretório raiz desse arquivo.
- O trabalho deverá ser enviado para caio.rns@gmail.com (com cópia para marcelolaureto@usp.br) com o assunto ACH2024-EP2. No corpo da mensagem e nas 1as linhas do código fonte deverá constar os nomes e números USP dos membros do grupo.

Não é necessário que os dois alunos do grupo enviem o código-fonte, basta um e-mail por grupo.

- Dúvidas a respeito das especificações ou a respeito da implementação do trabalho serão sanadas até o dia 08/08/2014. Dúvidas encaminhadas após este prazo serão ignoradas.
- Além da correção do programa, será considerada a qualidade da documentação do código fonte.
- Se houver evidência de plágio entre trabalhos de grupos distintos, os mesmos serão desconsiderados.