Centro Federal de Educação Tecnológica de Minas Gerais
Departamento de Computação
Laboratório de Algoritmos e Estruturas de Dados II
Prof. Thiago de Souza Rodrigues
2022-04-29

Prática 03 – Implementação do TAD ***ArvoreB***

- **Data de Entrega: 13/05/2022**
- **Deve ser entregue um relatório via moodle:**

  Nome

  Tabela com os Valores encontrados no item 3;

  Gráfico gerado no item 4;
- **Deve ser postado o projeto via moodle**

1) Implemente a classe ***Item***, como especificada abaixo para ser utilizada no T.A.D.;

```
package Item;
public class Item {
    private int chave;
    public Item(int chave) {
        this.chave = chave;
    }
    public int compara(Item it) {
        Item item = it;
        if (this.chave < item.chave)
            return -1;
         else if (this.chave > item.chave)
            return 1;
         return 0;
    }
    public int getChave() {
        return chave;
    }
}
```

2) Implemente uma classe ***ArvoreB*** para manipular uma árvore B de ordem definida pelo usuário, onde:

- cada página contenha objetos do tipo `Item`;
- método para realizar a inserção de elementos na árvore deve ser implementado;
- método para realizar a pesquisa de algum elemento ser passado por parâmetro deve ser implementado;

3) Utilizando a implementação já realizada do TAD ***ArvoreSBB*** e a implementação do TAD ***ArvoreB*** faça os seguintes experimentos:

a) gerar ***árvores SBB*** a partir de **n** elementos ***ORDENADOS***, com **n** variando de 10.000 até 100.000, com intervalo de 10.000.

   Em cada árvore gerada pesquisar por um elemento ***não existente*** e verificar o número de comparações realizadas na pesquisa em cada árvore;

b) gerar ***árvores B*** de ***ordem 2*** a partir de **n** elementos ***ORDENADOS***, com **n** variando de 10.000 até 100.000, com intervalo de 10.000.

   Em cada árvore B gerada pesquisar por um elemento ***não existente*** e verificar o ***número de comparações*** na pesquisa em cada árvore.

c) gerar ***árvores B*** de ***ordem 4*** a partir de **n** elementos ***ORDENADOS***, com **n** variando de 10.000 até 100.000, com intervalo de 10.000.

   Em cada árvore B gerada pesquisar por um elemento ***não existente*** e verificar o ***número de comparações*** na pesquisa em cada árvore.

d) gerar ***árvores B*** de ***ordem 6*** a partir de **n** elementos ***ORDENADOS***, com **n** variando de 10.000 até 100.000, com intervalo de 10.000.

   Em cada árvore B gerada pesquisar por um elemento ***não existente*** e verificar o ***número de comparações*** na pesquisa em cada árvore.

4) Fazer um único gráfico de **`n x número de comparações`** levando em consideração as árvores geradas. Explique o comportamento dos gráficos gerados;