

## Técnicas de Busca e Ordenação

### Roteiro de Laboratório 12– Árvores e recursão

OBS.: Todos os TADs pedidos estão na mesma pasta, mas a utilização dos mesmos foi feita dentro do TAD bst.

#### 1. TAD BST:

- 1.1 – EmptyBst();
- 1.2 – InsertBst();
- 1.3 – DestroyBst();

#### 2. Cliente do TAD BST:

- 2.1 – Que nesta seed para os numeros randomicos, os valores se distanciam da média, em média, 4.96
- 2.2 – Máximo é N, caso todos os nós sejam diferentes e sequencialmente maiores um do outro. O mínimo é quando a árvore é perfeitamente balanceada, então o mínimo é o somatório de 0 até 20 (para N sendo 1.000.000) de  $2^i$ .

#### 3. Caminhando em árvore:

- 3.1 – Funções genéricas nomeadas como: rec\_preorder(), rec\_inorder(), rec\_postorder().

#### 4. TAD Pilha:

- 4.1 Stack.c, Stack.h.

#### 5. Non-Recursive BST traversal:

- 5.1 Funções nomeadas como: stack\_preorder(), stack\_inorder(), stack\_postorder().

#### 6. Análise empírica:

- 6.1 – Os códigos com recursão foram mais rápidos que os iterativos. A depender do código e da complexidade, pode ser que seja mais fácil e rápido utilizar programas recursivos.

#### 7. TAD Fila:

- 7.1 Queue.c, Queue.h

#### 8. Non-Recursive level-order traversal:

- 8.1 Implementada.