Instituto de Ciência e Tecnologia - São José dos Campos, SP
UC: Algoritmos e Estrutura de Dados 2 - 2º semestre/2020
Prof.: Dra. Lilian Berton
Aluno: Davi Juliano Ferreira Alves - 133595

Atividade 7 - Prática

Entrega: 19/01/2020

Análise de complexidade do código para Árvore Binária de Busca, vetores ordenados e desordenados

A estrutura de Árvore Binária é dada por nós que contém informações, que no código foram ditas como “info”. Para a inserção, utilizamos as funções rand e as funções do pacote time.h. As configurações do vetor randômico, do vetor ordenado e das ABB’s são dadas pela tarefa anterior, ou seja, houve apenas a adição da estrutura de dados da AVL. A tabela do tempo pode ser observada abaixo:

	Inserir 10.000 elementos	Pesquisa elemento 50	Pesquisa elemento 5000
Vetor Randômico	0.00026 segundos	0.000026 segundos	0.000029 segundos
Vetor Ordenado (Pelo Insertion Sort)	0.117073 segundos	0.000001 segundos	0.000004 segundos
Árvore Binária de Busca 1	0.002527 segundos	0.000004 segundos	0.000002 segundos
Árvore Binária de Busca 2	0.205311 segundos	X	X
Árvore AVL	0.005061 segundos	0.0000017 segundos	0.000007 segundos

Figura 1 - Tabela dos Tempos de cada uma das ações pedidas.

```
O algoritmo do Insertion Sort demorou 0.117073 segundos para ordenar o vetor randômico
A inserção da ABB demorou 0.002527 segundos para inserir o vetor randômico
A inserção da ABB demorou 0.205311 segundos para inserir o vetor ordenado
O algoritmo de inserção demorou 0.005061 segundos para inserir na AVL
```

Figura 2 - O Output dos tempos de inserção.

```
O Busca Vetor demorou 0.000001 segundos para buscar 50 no vetor ordenado
O Busca Vetor demorou 0.000026 segundos para buscar 50 no vetor randômico
O Busca ABB demorou 0.000004 segundos para buscar 50 na Árvore Binária
Foi encontrado o valor 50!
O algoritmo de busca demorou 0.000017 segundos para buscar 50 na AVL
```

Figura 3 - O Output dos tempos de busca no valor 50 nas estruturas de dados

```
O Busca Vetor demorou 0.000029 segundos para buscar 5000 no vetor ordenado
O Busca Vetor demorou 0.000004 segundos para buscar 5000 no vetor randômico
O Busca ABB demorou 0.000002 segundos para buscar 5000 na Árvore Binária
Foi encontrado o valor 5000!
O algoritmo de busca demorou 0.000007 segundos para buscar 5000 na AVL
```

Figura 4 - O Output dos tempos de busca no valor 5000 nas estruturas de dados

Analisando a complexidade de cada uma das ações, temos que as complexidades das estruturas de dados anteriores já foram descritas no trabalho anterior, portanto será atentado ao que a Árvore AVL tem por complexidade. A inserção tem uma complexidade quase-linear (O(log(n))), pois ela busca na altura h a folha onde deve ser inserido e logo após faz o auto-balanceamento. Para as funções sobressalentes, como remoção e busca, a complexidade também segue o padrão logarítmico, sendo h o valor do logaritmo.

Como utilizamos a Árvore AVL, temos o resultado como:

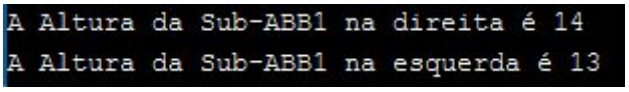

Figura 4 - O Output da altura das sub-árvores.

O código da altura das árvores é dado por uma complexidade O(h), sendo h a altura da árvore, pois é uma recursão que faz todas as averiguações da altura da árvore, ou seja, passa por todos os nós da árvore até chegar nas folhas. A função de mostrar as folhas das árvores possuem a mesma complexidade O(h), pois precisa mostrar toda e qualquer folha que exista nas ramificações da árvore.