Universidade Federal de Ouro Preto - UFOP

Estrutura de Dados I - BCC 202

TRABALHO PRÁTICO 03

Marcelo Edivan **Matrícula: 16.2.4040**

Ouro Preto, 27 de Agosto de 2017.

Problema Proposto

O problema proposto para ambos os algoritmos (árvore AVL e tabela Hash) foi o mesmo, levando em conta que ambos seriam eficientes, já que o problema se trata do cadastramento de alunos em um banco de dados onde o número de matrícula seria usado para fazer as manipulações importantes (inserir, remover, atualizar, pesquisar).

A árvore AVL organizaria essa matrícula de forma semelhante a árvore binaria com o diferencial de ser uma árvore balanceada tornando sua eficiência maior e sendo uma boa escolha para trabalhar em um sistema onde o principal critério de organização é o número da chave (matrícula) dos itens que na árvore serão armazenados.

Já a tabela Hash (com listas encadeadas) trabalharia organizando essas matrículas nos índices de um vetor, tratando as colisões com listas encadeadas, isso faz com que o acesso aos itens armazenados seja feito de forma mais rápida e com um custo menor.

O meu principal objetivo de estabelecer um mesmo problema para ambos os algoritmos é medir qual algoritmo se sairá melhor em relação ao tempo medindo seu tempo de execução com duas entradas diferentes, a primeira com pouco volume de dados e a segunda com um volume de dados maior. O resultado foi inconclusivo pois ambos os algoritmos tiverem resultados extremamente semelhantes sendo necessário uma massa gigantesca de dados (o que eu não tenho como fornecer) para analisar de fato se há uma diferença em tempo entre os dois algoritmos. Mas ao analisar as complexidades dos algoritmos acima citados espera-se que a árvore AVL tenha um resultado melhor se o número de colisões da tabela Hash for muito alto o que poderia ser sanado com uma boa proposta de correção de colisões fazendo assim, pelo menos a meu ver, da tabela Hash uma melhor resposta para se trabalhar com manipulação e organização de chaves (no caso matrícula).

Tabela de Complexidade

	Complexidade de tempo			Complexidade de memória	
	Árvore AVL	Tabela Hash		Árvore AVL	Tabela Hash
		Melhor caso	Pior Caso		
Inserir	O(log n)	O(1)	O(n)	O(1)	O(1)
Remover	O(log n)	O(1)	O(n)	O(1)	O(1)
Pesquisar	O(log n)	O(1)	O(n)	O(1)	O(1)