

Relatório CPU

Sistemas Digitais

Frederico Bulhões de Souza Ribeiro

11208440

André Niero Setti

10883901

Respostas

1. Na estrutura 2 em cada inserção temos que percorrer a lista inteira para chegar ao final, uma operação que leva $O(n)$ de tempo, onde n = número de elementos atualmente na lista.
Já caso pudéssemos utilizar um ponteiro que aponta para o final da lista teríamos uma inserção em $O(1)$ já que pode ser simplesmente feita com a mudança de alguns ponteiros.
2. A lista encadeada que insere na primeira posição tem o menor tempo de inserção, já que para inserir têm que somente mudar 3 ponteiros, uma operação $O(1)$. Já o maior tempo de inserção se refere à lista ordenada, pois embora seja contra-intuitivo (já que a lista que insere no final sempre vai percorrer mais do que a ordenada) é mais rápido para o computador percorrer a fila inteira do que ter que parar no elemento certo, pois temos menos decisões à se tomar, e as decisões são bem custosas para um processador realizar.
3. Para as listas encadeadas, temos os seguintes fatores que influenciam o tempo de execução:
 - a. Ordem dos valores inseridos
 - i. No caso da lista ordenada, caso inseríssemos os elementos de forma ordenada de maior para menor teríamos um tempo de execução similar ao da lista que insere na frente, pois cada inserção seria em $O(1)$
 - b. Quantidade de valores
 - i. Na lista encadeada que inserem no final e na lista ordenada temos que o tempo de inserção é proporcional ao número de elementos na lista atualmente. Logo caso mais elementos fossem inseridos, cada inserção seguinte levaria mais tempo.
 - ii. Em todas as listas o tempo de busca é proporcional à quantidade de elementos, já que tem que percorrer todos os elementos para achar o correto.
 - c. Tamanho dos valores

- i. Já o tamanho dos valores não importa para o tempo, já que o único momento em que o valor é usado durante a inserção e busca é em comparações, e o tamanho do número não influencia no tempo da comparação.
- 4. Não há diferença significativa nos tempos de busca observados na AVL e na árvore binária de busca. Isso se deve pelo fato de uma árvore binária de busca, caso os elementos sejam inseridos de forma aleatória, ter uma altura esperada de $O(\log n)$, na mesma classe da AVL. Porém em um caso menos especial a árvore binária de busca pode ter altura $O(n)$ e nesse caso teria tempo pior.
- 5. No caso dos valores de entrada fornecidos por esse projeto, de valores aleatórios, a árvore de busca binária tem uma altura esperada de $O(\log n)$, a mesma classe da AVL. Então nesse caso em especial a árvore de busca binária tem o mesmo tempo esperado que a AVL.
- 6. Dentre as estruturas implementadas no projeto a estrutura que é mais recomendada para busca e inserção é a AVL, já que é a única estrutura em que no pior caso temos inserção e busca não linear, em $O(\log n)$.
- 7. Durante esse projeto o maior desafio foi a implementação da AVL, e com isso aprendi sobre o funcionamento interno de árvores binárias e listas encadeadas, e também entendi melhor como a entrada pode alterar o tempo de execução.