

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Instituto de Informática

Cadeira: Estrutura de dados

Professora Viviane Moreira

Análise Comparativa de Desempenho: ABPs e AVLs

Leonardo Vianna Feiteira

Leonardo Boaventura Bombardelli

1. Descrição e introdução ao trabalho:

Esse relatório tem como objetivo dissertar sobre os resultados do trabalho final de Estrutura de Dados, cujo objetivo era implementar funções para inserção, consulta e remoção de nodos em Árvores Binárias de Pesquisa (ABPs) e na estrutura de árvores balanceadas AVL, comparando os resultados obtidos e verificando quais estruturas eram mais eficientes em determinados casos.

Foram implementadas em C funções para a inserção, consulta e remoção de nodos para ambas estruturas. O programa construído recebe como argumento dois arquivos: Um roteiro com o qual são especificadas as ações que o programa deve tomar (Inserção, remoção, consulta ou estatísticas sobre a árvore) e um arquivo de saída, onde essas estatísticas serão armazenadas.

2. Resultados obtidos:

Temos, a seguir, tabelas que mostram os resultados obtidos nas estatísticas do programa., e logo mais, dissertaremos sobre seus dados. Foram usados como arquivos de teste os arquivos disponíveis no Moodle da disciplina, tanto aqueles cujos valores estão organizados aleatoriamente quanto aqueles cujos valores são inseridos em ordem. Ambos os resultados dos testes serão apresentados aqui, visto que o comportamento das árvores difere bastante nos dois casos. Os arquivos de números aleatórios serão referenciados nesse relatório como “Caso aleatório”, e os ordenados, por sua vez, “Caso sorted”. Vale ressaltar que os valores de tempo são valores aproximados, o que explicam os resultados de 0 milissegundos que aparecem nas tabelas.

2.1. Tempo de processamento da ABP (Em milissegundos) - Caso aleatório

Número de nodos	100	1000	10000	100000
Inserção	0	1	9	265
Consulta	0	0	7	148
Remoção	0	0	8	146

2.2. Tempo de processamento da AVL (Em milissegundos) - Caso aleatório

Número de nodos	100	1000	10000	100000
Inserção	0	1	12	71

Consulta	0	0	8	41
Remoção	0	1	33	1720

2.3. Quantidade de comparações em ABP - Caso aleatório

Número de nodos	100	1000	10000	100000
Inserção	699	10301	153375	16793479
Consulta	2297	32903	480125	50580437
Remoção	2862	37830	534348	48520108

2.4. Quantidade de comparações em AVL - Caso aleatório

Número de nodos	100	1000	10000	100000
Inserção	961	14419	195145	2509449
Consulta	1673	26486	366989	4706423
Remoção	7813	160177	4137234	355575604

Omitiremos nessas tabelas os resultados de rotações e de fator das árvores. O primeiro será omitido por não existirem rotações em ABPs, e portanto, não temos nesse caso um meio de comparação entre as duas estruturas. O segundo, por sua vez, se deve ao fato de que fatores de árvores AVL oscilam, entre, no máximo, 1 e -1, fazendo dessa característica outra péssima forma para comparar o desempenho entre ABPs e AVLs.

Seguiremos, agora, para os resultados obtidos com os testes de inserção de valores em ordem. Esse teste é interessante de se analisar pois conseguimos resultados totalmente diferentes dos anteriores, questão que será discutida a fundo mais adiante.

2.5. Tempo de processamento da ABP (Em milissegundos) - Caso sorted

Número de nodos	100	1000	10000	100000
-----------------	-----	------	-------	--------

Inserção	0	9	414	61975
Consulta	0	5	231	25840
Remoção	0	1	2	19

2.6. Tempo de processamento da AVL (Em milissegundos) - Caso sorted

Número de nodos	100	1000	10000	100000
Inserção	0	1	4	49
Consulta	0	1	3	26
Remoção	0	1	7	91

2.7. Quantidade de comparações ABP - Caso sorted

Número de nodos	100	1000	10000	100000
Inserção	4950	499500	49995000	4999950000
Consulta	15050	1500500	150005000	15000050000
Remoção	997	9997	99997	999997

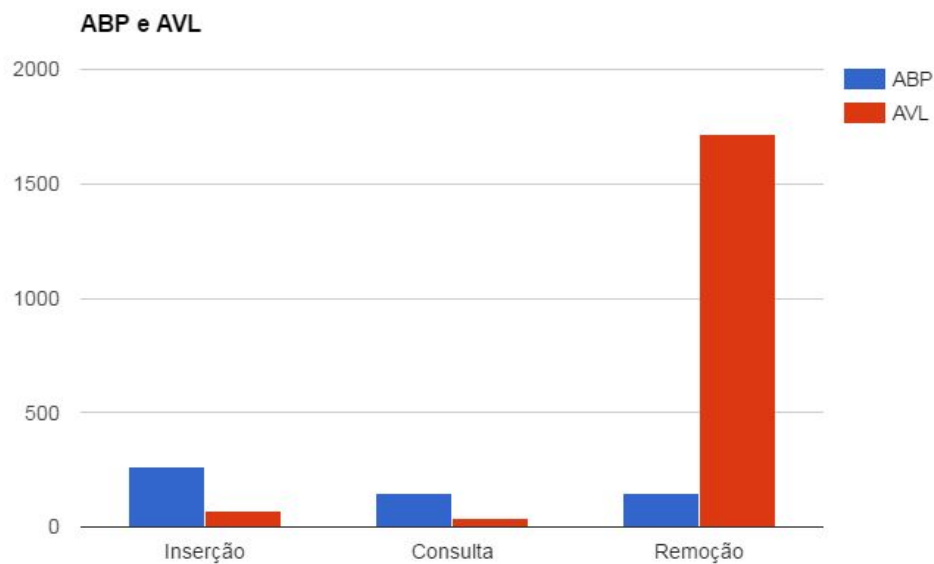
2.8. Quantidade de comparações AVL - Caso sorted

Número de nodos	100	1000	10000	100000
Inserção	859	11957	153589	1868895
Consulta	1640	25961	360893	4606838
Remoção	6135	110334	1599808	20780013

3. Conclusões acerca dos resultados obtidos e considerações finais:

Podemos perceber nítidas diferenças de desempenho das duas estruturas, além de diferenças de desempenho das mesmas dependendo da ordem nas quais os nodos são inseridos. Começamos fazendo uma comparação de tempos de inserção, consulta e remoção das duas árvores no caso aleatório.

3.1. Comparação entre ABP e AVL



Podemos perceber que para fazer consultas, a AVL é mais eficiente que a ABP. Isso se deve ao fato de que, como a AVL é uma árvore balanceada e a consulta depende da altura da árvore, consultas nessa estrutura serão muito mais eficientes. Por outro lado, o tempo de remoção em AVLs é muito maior que o tempo de remoção em ABPs. Isso, também, se deve ao fato de que, para cada remoção de nodo que ocorre, a árvore deve garantir que está balanceada, e portanto, deverá fazer inúmeras rotações para garantir esse balanceamento.

Podemos perceber essa discrepância entre as duas estruturas em consultas de nodos observando esse gráfico de tempo de consulta de ABPs e AVLs em comparação ao tempo:

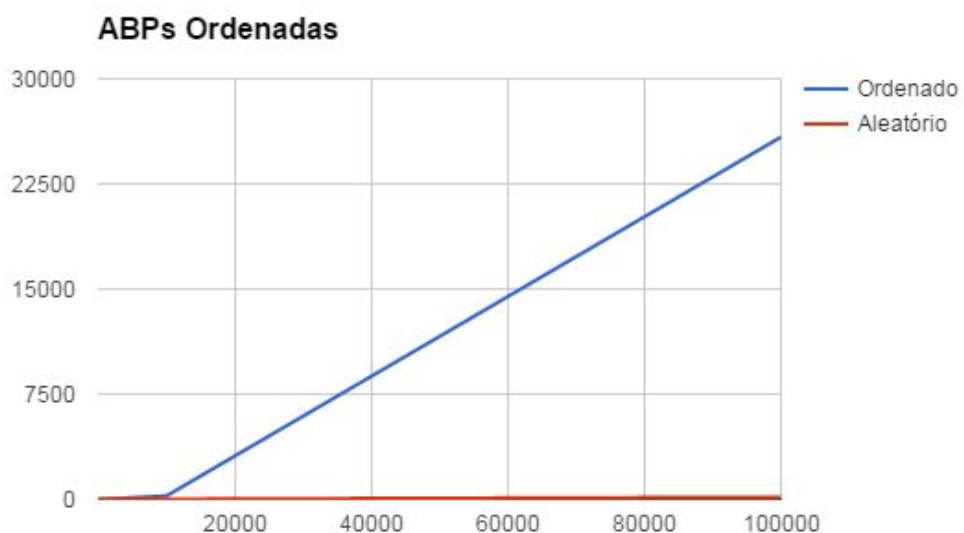
3.2. Comparação entre tempo de consulta ABP e AVL



Embora tenhamos gráficos aparentemente lineares, vale salientar que o mesmo foi feito a partir de uma baixa diferença entre número de nós. Essa baixa amostra, por sua vez, faz o gráfico ser linear. Porém, percebemos, mesmo nesse gráfico, um crescimento exponencial de consultas no caso das ABPs, enquanto as AVLs se mantêm mais constantes quando a quantidade de nós a consultar aumenta. Isso se deve, novamente, ao fato de que as AVLs são balanceadas, e portanto, tem um tempo de consulta médio menor.

Podemos perceber, por fim, uma grande discrepância das ABPs para as AVLs quando inserimos os valores em ordem. Isso se deve ao fato de que uma ABP, sem rotações para balancear a árvore, acaba regredindo para uma estrutura de lista quando inserimos os valores em ordem.

3.3. Comparação entre tempo de consulta em ABPs aleatório e sorted



Analizadas, por fim, esses casos interessantes de discrepâncias entre ABPs e AVLs, podemos, enfim, fazer análises acerca de seus desempenhos para determinados casos. As AVLs se mantiveram constantes em seus tempos de consulta, mas tem um custo computacional maior para sua inserção e, principalmente, remoção de nodos. ABPs, por outro lado, apresenta resultados de consulta