Relatório de Desempenho de Árvores Binárias e Árvores AVL

Lucas Azevedo Dias

Este relatório analisa o desempenho das operações de inserção, remoção e busca em uma árvore binária e em uma árvore AVL (Árvore Binária de Busca Balanceada) com diferentes tamanhos de entrada. As operações foram realizadas em ambas as estruturas de dados e foram cronometradas para avaliar o tempo de execução.

Implementação e Metodologia

Foram implementadas duas classes principais: BinaryTree para a árvore binária e BinaryTreeAVL para a árvore AVL. Cada classe possui métodos para inserção, remoção e busca de valores inteiros.

O programa de teste executou as seguintes operações:

- **Inserção de Valores Aleatórios:** Foram gerados valores inteiros aleatórios e inseridos nas árvores binária e AVL. O tempo de execução foi registrado.
- **Remoção de Valores Inseridos:** Valores anteriormente inseridos foram removidos das árvores binária e AVL. O tempo de execução foi registrado.
- **Busca de Valores Inseridos:** Valores inseridos foram buscados nas árvores binária e AVL. O tempo de execução foi registrado.

Resultados

Os testes foram executados com cinco diferentes tamanhos de entrada: 100, 500, 1000, 10000 e 20000 valores inteiros aleatórios. Abaixo estão os resultados do tempo de execução em milissegundos (ms) para cada operação e cada tipo de árvore.

Operações em Árvore Binária (sem AVL):

1. Inserção:
   - 100 valores: 1ms
   - 500 valores: 0ms
   - 1000 valores: 1ms
   - 10000 valores: 3ms
   - 20000 valores: 7ms

2. Remoção:
   - 100 valores: 0ms
   - 500 valores: 1ms
   - 1000 valores: 0ms
   - 10000 valores: 4ms

- 20000 valores: 6ms

3. Busca:
   - 100 valores: 0ms
   - 500 valores: 0ms
   - 1000 valores: 5ms
   - 10000 valores: 1ms
   - 20000 valores: 3ms

Operações em Árvore AVL:

1. Inserção:
   - 100 valores: 2ms
   - 500 valores: 3ms
   - 1000 valores: 4ms
   - 10000 valores: 787ms
   - 20000 valores: 4045ms

2. Remoção:
   - 100 valores: 1ms
   - 500 valores: 1ms
   - 1000 valores: 5ms
   - 10000 valores: 674ms
   - 20000 valores: 3932ms

3. Busca:
   - 100 valores: 1ms
   - 500 valores: 0ms
   - 1000 valores: 0ms
   - 10000 valores: 1ms
   - 20000 valores: 2ms

Discussão dos Resultados

Os resultados mostram que a árvore AVL tende a ter um desempenho melhor em operações de busca em comparação com a árvore binária, uma vez que a árvore AVL é balanceada e mantém a altura da árvore em níveis controlados. No entanto, a árvore AVL pode ser um pouco mais lenta nas operações de inserção e remoção, devido à necessidade de reequilibrar a árvore após cada operação.

O desempenho das operações em ambas as estruturas de dados piora à medida que o número de valores aumenta, como esperado. A complexidade temporal das operações em árvores binárias não balanceadas pode levar a um aumento significativo no tempo de execução à

medida que a entrada cresce, enquanto a árvore AVL consegue manter um desempenho mais consistente.

Em resumo, a escolha entre uma árvore binária e uma árvore AVL depende das operações que você precisa realizar com mais frequência. Se as buscas são predominantes e os dados são frequentemente atualizados, a árvore AVL pode ser uma escolha melhor devido ao seu desempenho mais estável. No entanto, se as inserções e remoções são mais comuns e a árvore não precisa ser frequentemente pesquisada, uma árvore binária pode ser mais adequada.

Conclusão

Este relatório demonstra o desempenho de operações em árvores binárias e árvores AVL com diferentes tamanhos de entrada. A escolha entre essas estruturas de dados depende das necessidades específicas do seu aplicativo e dos tipos de operações que serão executadas com mais frequência. A árvore AVL oferece um desempenho mais previsível em operações de busca, enquanto a árvore binária pode ser mais eficiente em cenários com foco em inserção e remoção.