UNIVERSIDADE FEDERAL DE MATO GROSSO

Universidade Federal de Mato Grosso
ICET - Instituto de Ciências Exatas e da Terra
Departamento de Ciência da Computação
Graduação em Ciência da Computação

Gabriel Nunes
Raissa Cavalcanti

Tarefa 01 - Árvores Binárias Balanceadas

Estrutura de Dados II

Barra do Garças
05 de Agosto 2024

Gabriel Nunes
Raissa Cavalcanti

Árvores Binárias Balanceadas

Relatório referente a Tarefa 01 da disciplina de Estrutura de Dados

Orientador: Professor Robson Lopes

Barra do Garças
05 de agosto de 2024

Introdução

A estrutura de dados conhecida como árvore binária desempenha um papel crucial em diversas aplicações computacionais, principalmente no que diz respeito à organização e manipulação eficiente de grandes volumes de dados. Dentre as variações de árvores binárias, destacam-se a Árvore Binária de Busca (ABB), a Árvore AVL e a Árvore Rubro-Negra, cada uma com suas particularidades em termos de balanceamento e desempenho. Este trabalho tem como objetivo implementar e analisar essas três estruturas de dados, utilizando um conjunto de valores inteiros gerados aleatoriamente. A análise comparativa se concentrará na altura das árvores, no número de rotações necessárias para o balanceamento e no desempenho das operações de busca, com o intuito de compreender as vantagens e desvantagens de cada abordagem.

Detalhamento

Etapa 1: Geração dos Dados

O primeiro passo deste projeto consiste na elaboração de um algoritmo capaz de gerar um arquivo contendo valores inteiros aleatórios, não repetidos, no intervalo de 1 a 100.000. O algoritmo receberá dois parâmetros: o número de valores a serem gerados e o nome do arquivo de saída. A geração dos dados será realizada de forma que cada valor seja armazenado em uma linha do arquivo, garantindo a unicidade e a aleatoriedade dos números.

Etapa 2: Implementação das Árvores

Nesta etapa, serão implementadas as funções necessárias para a construção e manipulação das três árvores binárias: Árvore Binária de Busca (ABB), Árvore AVL e Árvore Rubro-Negra. As funções incluirão operações de inserção e busca, bem como a análise das estruturas resultantes. Especificamente, serão implementadas funções para determinar a altura das árvores, contabilizar o tempo e o número de comparações durante as buscas, e registrar o número de rotações realizadas durante as inserções nas árvores balanceadas.

Detalhamento

Etapa 3: Geração e Carregamento dos Arquivos de Entrada
Serão gerados três arquivos de entrada contendo 5.000 números e três arquivos com 20.000 números, utilizando o algoritmo desenvolvido na Etapa 1. Cada um desses arquivos será carregado nas três estruturas de árvores (ABB, AVL e Rubro-Negra). Durante esse processo, serão registrados a altura final de cada árvore e o número total de rotações realizadas. Esta etapa permitirá uma análise comparativa da eficiência estrutural das três abordagens em termos de balanceamento e altura das árvores.

Etapa 4: Análise de Desempenho nas Operações de Busca
Para cada um dos arquivos gerados na Etapa 3 e carregados nas árvores, será criada uma função que selecionará aleatoriamente 20% dos valores gerados e realizará buscas por esses valores nas árvores. Serão medidos o tempo de execução e o número de comparações realizadas para cada operação de busca, proporcionando uma avaliação detalhada do desempenho das três estruturas de dados em operações de busca frequentes.

Processo

Geração de Dados: O objetivo desta etapa foi gerar arquivos contendo números inteiros únicos no intervalo de 1 a 100.000, com duas quantidades de valores: 5.000 e 20.000. Para isso, foi desenvolvido o código 'gerar_numeros.c.'' A execução do código é realizada da seguinte forma:

1. Compilar o código: gcc gerar_numeros.c -o gerar_numeros
2. Gerar arquivos com 5.000 números: ./gerar_numeros 5000 arquivo_5000.txt
3. Gerar arquivos com 20.000 números: ./gerar_numeros 20000 arquivo_20000.txt

Processo

Implementação das Árvores: Implementamos as funções necessárias para construção, inserção, busca e análise de cada tipo de árvore binária. As árvores implementadas são:

1. **Árvore Binária de Busca (ABB):**
   - Código: abb.c
   - Funções: Inserção, busca, cálculo da altura, e percurso em ordem.
   - Contagem de Rotações: Não aplicável.
2. **Árvore AVL:**
   - Código: avl.c
   - Funções: Inserção com balanceamento, busca, cálculo da altura, e percurso em ordem.
   - Contagem de Rotações: Implementada.
3. **Árvore Rubro-Negra (ARN):**
   - Código: arn.c
   - Funções: Inserção com balanceamento, busca, cálculo da altura, e percurso em ordem.
   - Contagem de Rotações: Implementada.

Análise de Desempenho: Carregamos os arquivos gerados na primeira etapa em cada tipo de árvore e contabilizamos a altura das árvores e o número de rotações executadas durante a inserção dos valores.

Avaliação de Busca: Nesta etapa, selecionamos aleatoriamente 20% dos valores gerados, buscamos cada valor na árvore e calculamos o tempo e o número de comparações realizadas.

Análise dos Dados

Os resultados obtidos a partir da execução dos algoritmos de Árvore Binária de Busca (ABB), Árvore AVL e Árvore Rubro-Negra (ARN) nos arquivos gerados fornecem uma visão detalhada do desempenho de cada estrutura de dados em termos de rotações, comparações e tempo de busca.

Resumo dos Resultados

Para os arquivos de 5.000 valores:

- **Árvore AVL:**
  - Número total de rotações: variou entre 3.487 e 3.549.
  - Número total de comparações: variou entre 11.511 e 11.523.
  - Tempo total de busca: variou entre 0.000981 e 0.001199 segundos.
- **Árvore Rubro-Negra (ARN):**
  - Número total de rotações: variou entre 2.929 e 2.999.
  - Número total de comparações: variou entre 11.539 e 11.715.
  - Tempo total de busca: variou entre 0.001061 e 0.001186 segundos.
- **Árvore Binária de Busca (ABB):**
  - Número total de comparações: variou entre 15.499 e 15.926.
  - Tempo total de busca: variou entre 0.000963 e 0.001212 segundos

.

Para os arquivos de 20.000 valores:

- **Árvore AVL:**
  - Número total de rotações: variou entre 13.831 e 13.905.
  - Número total de comparações: variou entre 54.067 e 54.396.
  - Tempo total de busca: variou entre 0.004543 e 0.005179 segundos.
- **Árvore Rubro-Negra (ARN):**
  - Número total de rotações: variou entre 11.595 e 13.831.
  - Número total de comparações: variou entre 54.395 e 54.580.
  - Tempo total de busca: variou entre 0.004338 e 0.004543 segundos.
- **Árvore Binária de Busca (ABB):**
  - Número total de comparações: variou entre 14.573 e 14.882.
  - Tempo total de busca: variou entre 0.001047 e 0.001124 segundos.

Análise dos Dados

1. **Rotações:**

- As árvores AVL executaram mais rotações em comparação com as ARN, refletindo o esforço adicional necessário para manter o balanceamento rigoroso característico das AVL.
- As ARN, embora também realizem rotações, necessitam de menos rotações em comparação com as AVL, resultando em um menor número total de rotações.

2. **Comparações:**

- A Árvore Binária de Busca (ABB) apresentou o maior número de comparações, o que é esperado devido à falta de balanceamento das ABB.
- As árvores AVL e ARN apresentaram um número de comparações similar, com as AVL geralmente apresentando um número ligeiramente menor de comparações.

3. **Tempo de Busca:**

- O tempo de busca foi consistentemente menor nas árvores AVL e ARN em comparação com a ABB para arquivos de 5.000 valores.
- Para arquivos de 20.000 valores, embora o tempo de busca das AVL e ARN tenha sido maior do que nos arquivos menores, ainda se manteve competitivo e, em alguns casos, menor do que o tempo de busca na ABB.

Considerações Finais

A análise dos resultados permite concluir que as árvores balanceadas (AVL e Rubro-Negra) oferecem uma performance superior em termos de busca e balanceamento em comparação com a Árvore Binária de Busca (ABB). A Árvore AVL, devido ao seu balanceamento estrito, realiza mais rotações do que a Rubro-Negra, mas essa característica também contribui para um menor número de comparações e tempos de busca mais competitivos.
A Árvore Rubro-Negra, embora execute menos rotações que a AVL, apresenta desempenho de busca e número de comparações muito próximos, o que a torna uma alternativa eficiente em cenários onde um menor número de rotações é desejado.

A análise das alturas das árvores confirma que as árvores balanceadas, AVL e Rubro-Negra, são superiores em termos de eficiência estrutural comparadas às ABBs. A Árvore AVL mantém alturas menores de forma consistente, garantindo excelente desempenho em operações de busca e inserção através de um balanceamento rigoroso. A Árvore Rubro-Negra, embora permita um pouco mais de variação na altura, oferece um bom equilíbrio entre altura e número de rotações necessárias para manter o balanceamento, resultando em uma performance robusta.
Por outro lado, as ABBs, devido à falta de um mecanismo de balanceamento, apresentam alturas significativamente maiores, tornando-as menos eficientes para operações com grandes conjuntos de dados. Para aplicações que requerem desempenho consistente e eficiente, especialmente com grandes volumes de dados, as árvores balanceadas como AVL e Rubro-Negra são as escolhas mais adequadas.

Por fim, a Árvore Binária de Busca (ABB), apesar de sua simplicidade de implementação, apresenta o pior desempenho em termos de número de comparações e, em muitos casos, no tempo de busca, devido à falta de balanceamento. Portanto, em aplicações onde o desempenho é crítico, especialmente com grandes conjuntos de dados, as árvores balanceadas, como a AVL e a Rubro-Negra, são preferíveis.