UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS – PICOS
Disciplina: Estrutura de Dados II
Curso: Bacharelado em Sistemas de Informação
Aluno: Aleff Ericlys Lustosa Araujo

Resumo do Projeto

O Relatório técnico apresenta a descrição de todos os métodos e conceitos utilizados para a resolução da primeira avaliação de Estrutura de Dados II.

1. Introdução

As árvores, são uma das mais importantes estruturas de dados não lineares. Diferente das listas nas árvores os dados estão dispostos de forma hierárquica. São estruturas eficientes e simples em relação ao tratamento computacional, há inúmeros problemas no mundo real que podem ser modelados e resolvidos através das árvores. Estruturas de pastas de um sistema operacional, interfaces gráficas, bancos de dados e sites da Internet são exemplos de aplicações de árvores. Esse relatório técnico foi organizado da seguinte maneira. Na Seção 2 serão apresentados alguns conceitos básicos dos assuntos abordados. A Seção 3 contém a metodologia implementada para a realização dos experimentos. Na Seção 4 apresenta os resultados da execução do programa e na Seção 5 contém a conclusão dos resultados alcançados. Na Seção 6 apêndice e na Seção 7 as referências usadas na produção deste relatório.

2. Seções Específicas

2.1. Estrutura de Árvore

As estruturas de dados de árvores são estruturas nomeadas assim por apresentarem um aspecto representativo semelhante às árvores da vida real [Cormen et al. 2002]. Diferente das estruturas de listas encadeadas, aqui as informações não são dispostas em forma sequencial, nas árvores os dados inseridos são armazenados de forma hierárquica. Nesse tipo de estrutura temos elementos inseridos que são armazenados em nós encadeados, cada nó de uma árvore binária aponta para até dois outros nós [Celes et al. 2004]. O nó que inicia a árvore e aponta para outros nós é denominado de raiz, a partir da raiz é possível chegar em qualquer outro nó da árvore.

2.2. Árvore Binária

A árvore binária de busca é uma estrutura de dados que adota a mecânica de hierarquias entre os elementos inseridos. Por se tratar de uma árvore binária, cada nó possui no máximo dois filhos. O fator diferencial dessa estrutura está na forma como os elementos são inseridos e organizados dentro da árvore, exemplifica a formação de uma árvore binária de Busca. Durante a inserção de um novo elemento, o valor do novo elemento é comparado ao valor do nó atual, para então decidir a direção onde esse novo elemento será inserido. É adotado como um padrão de programação que valores maiores que o nó atual são inseridos à direita do nó, enquanto que valores menores são inseridos à

esquerda. Dessa forma, a árvore binária de busca segue esse padrão em que todos os valores menores que a raiz estão à esquerda, e todos os valores maiores que a raízes tão inseridos a direita. A árvore binária de busca proporciona rapidez e eficiência no tempo de busca de um dado, é uma diferença significativa na diminuição de tempo e custo computacional em comparação a estruturas de listas encadeadas.

2.3. Árvore AVL

Uma árvore AVL nada mais é que uma estrutura de dados com árvore de forma balanceada. O termo balanceada significa dizer que a diferença na altura das sub árvores esquerda e direita não é maior que 1 e nem menor que -1. Dessa forma, garantindo uma maior estabilidade para realizar operações dentro da árvore. A Figura 2 exemplifica a formação de uma árvore AVL. Uma árvore balanceada garante um ganho de performance para as diferentes operações que serão realizadas na estrutura de dados, tais como busca, inserção e remoção.

3. Metodologia

Nessa seção apresenta a metodologia e ambiente utilizados para realizar a análise e comparação das estruturas de dados citadas. Durante a fase de testes o ambiente que foi utilizado, consiste em um computador para a execução de algoritmos e medição de dados. Todos os arquivos foram executados por um compilador web. As configurações de hardware do dispositivos̃ ao apresentadas na Tabela 1.

Dispositivo	Processador	Memória RAM	Sistema Operacional
Notebook Acer A515-51G-C690	Intel I7-8550U	8GB	Ubuntu 20.04.4

Tabela 1. Configurações de Hardware

3.1. Funcionalidades Utilizadas

- Funções implementadas nos algoritmos das questões 1 e 2 :

  novoNo - recebe um valor inteiro e retorna uma estrutura de tipo arvNum;
  insereNum - Recebe a raiz da árvore e um ponteiro para uma estrutura do tipo arvNum, e depois insere essa estrutura na posição correta da árvore;
  mostraArv - Mostra todos os valores contidos na árvore;
  busca - Recebe um valor e a raiz da árvore e verifica se o mesmo existe na árvore;
  limpaArv - Liberar a memória de toda a árvore;

- Funções implementadas nas questões 2 e 4, referentes a rotação e balanceamento da árvore AVL:

ehFolha - Recebe um nó da árvore e retorna 1 se ele for uma folha e 0 caso contrário;
profundidade - Recebe a raiz da árvore e o endereço de memória de outras duas variáveis de tipo inteiro e retorna a maior e a menor profundidade da árvore;
balance - recebe o nó e faz os testes de balanceamento e chama as funções de rotação se necessário;
balanceHor - Rotação simples no sentido horário;
balanceAntH - Rotação simples no sentido horário.

Funções auxiliares para o balanceamento da árvore AVL
fatBalance - Retorna o fator de balanceamento da árvore;
altura - Recebe a raiz ou um nó da árvore e devolve a altura;

OBS: as funções citadas acima aparecem repetidas 3 vezes, uma para cada estrutura de dados

- Funções implementadas nos algoritmos das questões 3 e 4:
  novoTermo - recebe um termo e retorna uma estrutura de tipo termos referente à raiz da árvore;
  cadTermo - Recebe a raiz da árvore e um ponteiro para uma estrutura do tipo termos, e depois insere essa estrutura na posição correta da árvore;
  mostra - Mostra todos os valores contidos na árvore;
  limpa - Liberar a memória de toda a árvore;

  dadosSubt - Gerencia a criação e inserção de um nó na árvore de subtermos
  novoSubt - recebe um termo e retorna uma estrutura de tipo subt referente à raiz da árvore;
  cadSubt - Recebe a raiz da árvore e um ponteiro para uma estrutura do tipo subt, e depois insere essa estrutura na posição correta da árvore;
  mostraT - Mostra todos os valores contidos na árvore;
  limpaSubt - Liberar a memória de toda a árvore;

  dadosInt - Gerencia a criação e inserção de um nó na árvore de páginas
  novoNo - recebe um termo e retorna uma estrutura de arvint subt referente à raiz da árvore;
  insereNum - Recebe a raiz da árvore e um ponteiro para uma estrutura do tipo arvint, e depois insere essa estrutura na posição correta da árvore;
  mostraP - Mostra todos os valores contidos na árvore;
  limpaPag - Liberar a memória de toda a árvore;

4. Resultados da Execução do Programa

Nesta seção, são apresentados os resultados utilizados para realizar a análise e comparação das estruturas de dados referente a árvore binária e a árvore AVL. Com o objetivo de identificar qual método será mais eficiente. Ao longo do experimento foram executados quatro algoritmos, dois referentes a inserção e busca de inteiros e os outros dois referente a inserção e busca de um árvore de strings, cada um com sua variação referente a árvore binária e a árvore AVL.

4.1 Programa de inserção e Busca de Inteiros

Todos os valores de tempo de inserção e tempo de busca que compõem a tabela 2, foram representados em nanosegundos, o tempo de busca é uma média de tempo de 5 buscas usando os valores (16224, 12659, 12499, 2379, 11706) em cada uma das árvores, tanto binária como AVL.

Os valores inseridos na árvore foram sorteados de forma randômica utilizando duas funções “rand()” e multiplicando o valor do primeiro com o segundo.

Os valores dos tempos de inserção e busta estão listados na Tabela 2 abaixo.

	ÁrvoreBinária			ÁrvoreAVL		
Número da Árvore	tempo de inserção	média do tempo de 5 buscas	diferença entre a maior e menor profundidade	tempo de inserção	média do tempo de 5 buscas	diferença entre a maior e menor profundidade
1	752209	701	15	1730753	650	4
2	697072	681	17	1566356	631	4
3	710440	972	13	1525857	549	4
4	620738	607	17	1556698	645	4
5	641655	708	16	1806354	814	4
6	245256	196	17	1750824	566	4
7	201732	229	20	1796473	667	3
8	217941	188	18	1742433	620	4
9	196985	223	15	1743233	577	4
10	206885	207	19	1689162	643	4
11	202318	196	13	1714085	557	4
12	215596	205	21	1672124	371	4
13	205046	179	16	1078519	427	3
14	214688	261	15	1098650	508	3
15	212301	206	13	1066942	364	4
16	207384	199	13	1055590	427	4
17	285693	300	16	807821	300	4
18	283479	272	14	781211	241	2
19	272399	304	15	791629	269	4
20	258135	290	19	780905	350	4
21	208619	214	19	776110	288	4
22	225820	241	17	709556	223	4

23	221470	271	14	604998	223	3
24	251146	289	19	606842	211	4
25	209424	227	13	602979	231	4
26	229183	278	16	780400	358	4
27	207691	233	17	749141	330	4
28	220281	229	14	720368	293	4
29	220479	220	18	641485	268	4
30	216212	226	15	610917	184	4

Tabela 2. Média da variação de tempo

4.2 Programa de árvore de termos, seus subtermos e suas respectivas páginas

Nas árvores de termos foram definidos termos e seus respectivos subtermos enquanto a inserção das árvores de páginas foram feitas de forma randômica. Os termos e subtermo utilizados não foram inseridos de forma alfabética. As inserções das duas árvores foram feitas na mesma ordem.

Os termos e subtermos usados foram:

- Aloca
  - Alocacao
  - Alocar
  - Aloquei
  - Alocam
  - Alocando
- Blinda
  - Blindar
  - Blindagem
  - Blindasse
  - Blindava
  - Blindam
- Cava
  - Cavar
  - Cavaram
  - Cavacao
  - Cavo
  - Cavei
- Desconsidera
  - Desconsiderar
  - Desconsideracao
  - Desconsiderei
  - Desconsiderava
  - Desconsideravam
- Estuda
  - Estudar

    - Estudei
    - Estudava
    - Estudavam
    - Estudare

- Forja
    - Forjar
    - Forjei
    - Forjavam
    - Forjaria
    - Forjava

- Gasta
    - Gastar
    - Gastarei
    - Gastava
    - Gastaria
    - Gastaram

Os valores dos tempos de inserção estão listados na Tabela 3 abaixo.

	Tempos de inserção de cada termo e seus subtermos na árvore	
Temos	Árvore Binária	Árvore AVL
Estudar	408	707
Cavar	333	779
Gastar	425	1031
Alocar	372	1246
Blindar	526	2564
Forjar	415	1250
Desconsiderar	229	1327

Tabela 3. Tempos de inserção

5. Conclusão

Esse relatório técnico apresentou uma análise das estruturas de dados baseadas em árvores binárias de busca e AVL, com o objetivo de comparar a eficiência das estruturas no cenários apresentados. Durante a primeira etapa foi observado que o tempo médio de inserção de valores na árvore binária é menor que o tempo de inserção da árvore AVL.

Em relação ao tempo de busca em ambas as árvores, em algumas ocasiões, ocorre de o tempo de busca na árvore binária ser menor, mas levando em consideração a média de todos os tempos de busca da árvore binária e todos os tempos de busca da árvore AVL, a árvore AVL tem uma eficiência maior. Essa eficiência se torna cada vez maior de acordo com o aumento do tamanho da árvore.

6. Apêndice

Todo código-fonte apresentado para as soluções das questões propostas seguem em anexo com a documentação.

7. Referências

Celes, W., Cerqueira, R., and Rangel, J. L. (2004). Introdução a estruturas de dados.Editora Campus.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2002). Algoritmos: teoria e prática.Editora Campus, 2:296.