UNIVERSIDADE FEDERAL DE VIÇOSA - CAMPUS FLORESTAL

Trabalho prático de Algoritmo e Estrutura de Dados 2

Nomes: Aline Cristina, Luana Tavares, Rafaella Ferreira
Matrículas: 5791, 5364, 5363

Florestal - MG
2024

UFV

Sumário

UFV

1. Introdução

O trabalho prático 1 da disciplina de Algoritmos e Estruturas de Dados 2 tem como objetivo de implementar e comparar duas técnicas de indexação e busca em grandes conjuntos de dados textuais: a árvore Patricia e a tabela hash. Estes algoritmos são amplamente utilizados para a organização e busca eficiente de dados, sendo essenciais em diversas aplicações computacionais.

O objetivo principal deste trabalho é desenvolver e avaliar a implementação de uma árvore PATRICIA e uma tabela HASH para armazenar e manipular dados textuais. A árvore PATRICIA é uma estrutura de dados compacta usada para a representação eficiente de conjuntos de strings, permitindo buscas e inserções rápidas. Por outro lado, a tabela HASH é uma estrutura que utiliza uma função de hash para distribuir dados de forma uniforme, proporcionando acesso rápido e eficiente.

O trabalho envolve a aplicação desses algoritmos em um contexto prático, onde serão utilizados para construir um índice invertido. Este índice é uma estrutura que mapeia palavras para a lista de documentos que as contêm, facilitando a recuperação de informações em sistemas de busca e análise de texto.

2. Metodologia

2.1. Divisão de Tarefas:

- **Estrutura Hash:** Aline Cristina assumiu a responsabilidade de implementar a estrutura básica da tabela hash. Este componente foi projetado para armazenar e gerenciar palavras de forma eficiente, utilizando funções essenciais como criação, inserção e gerenciamento da tabela.
- **Índice Hash:** Para a construção do índice invertido utilizando a estrutura hash, Aline Cristina, Luana Tavares e Rafaella Pinheiro trabalharam em conjunto. A colaboração envolveu a definição da estrutura dos índices e a integração com a tabela hash, garantindo que as palavras fossem indexadas como solicitado na especificação do trabalho prático.
- **Busca Hash:** Luana Tavares foi encarregada de desenvolver a funcionalidade de busca na tabela hash. Esta parte do trabalho incluiu a criação de algoritmos para consultar a tabela e recuperar informações rapidamente.
- **Estrutura Patricia:** Rafaella Pinheiro focou na implementação da árvore Patricia. O trabalho incluiu a criação e manipulação dos nós da árvore, projetados para armazenar palavras de forma eficiente e otimizar as operações de busca.
- **Índice Patrícia:** A construção do índice invertido usando a árvore Patricia foi uma tarefa compartilhada entre Aline Cristina, Luana Tavares e Rafaella Pinheiro. O grupo trabalhou na definição dos formatos de dados e na integração com a árvore para garantir uma indexação precisa.
- **Busca Patricia:** Luana Tavares desenvolveu a funcionalidade de busca na árvore Patricia, implementando algoritmos para realizar consultas e recuperar dados de maneira eficiente.
- **Comparações:** Aline Cristina foi responsável por realizar as comparações entre as abordagens hash e Patrícia. Essa análise envolveu avaliar o desempenho em termos de comparação de busca e inserção em ambas as estruturas.
- **Documentação**: Aline Cristina e Rafaella Pinheiro foram responsáveis por elaborar a documentação do projeto.

3. Resultados do desenvolvimento

3.1. Detalhes Técnicos de Implementação

3.1.1. Implementação da Árvore Patricia

A árvore Patrícia**(Practical Algorithm to Retrieve Information Coded in Alphanumeric),** foi implementada para armazenar e gerenciar os índices invertidos dos termos. A seguir, detalhamos os principais aspectos da sua implementação:

- **Estrutura da Árvore**:
  - **Nó Interno**: Cada nó interno contém um caractere que ajuda a decidir a direção de inserção e busca. A comparação do caractere determina se o próximo nó a ser visitado é o da esquerda ou da direita.
  - **Nó Folha**: Armazena as palavras do índice invertido, associadas a uma lista de ocorrências. Cada ocorrência contém o ID do documento e a posição do termo dentro do documento.
  - **Nó Externo**: Armazena a palavra completa.
- **Inserção**: A inserção de novos termos na árvore Patrícia é feita comparando-se os caracteres dos termos com os caracteres armazenados nos nós. Se o termo não for encontrado, novos nós são criados conforme necessário para armazenar o termo e suas ocorrências.
- **Busca**

1. **Início da Busca**
   - A busca começa na raiz da árvore. Se a árvore estiver vazia, a busca retorna um valor indicando que a palavra não foi encontrada.
2. **Verificação do Tipo de Nó**
   - Se o nó atual for um nó externo, a busca verifica se a palavra desejada corresponde à palavra armazenada nesse nó. Se houver uma correspondência, o ponteiro para a palavra armazenada é retornado. Se não houver correspondência, a busca retorna um valor indicando que a palavra não foi encontrada.
3. **Busca em Nós Internos**
   - Se o nó atual for um nó interno, a busca prossegue para um dos filhos do nó. A decisão de qual filho seguir é baseada na comparação de um caractere específico da palavra com um caractere armazenado no nó.
   - **Comparação dos Caracteres**
     - O caractere na posição da palavra é comparado com o caractere armazenado no nó. Se o caractere da palavra corresponde ao caractere do nó, a busca continua no filho direito; caso contrário, a busca continua no filho esquerdo.

3.1.2. Implementação da Tabela Hash

A Tabela Hash foi projetada para fornecer uma estrutura eficiente para armazenamento e recuperação de dados. Detalhamos os principais aspectos da sua implementação:

- **Estrutura da Tabela**: A tabela hash utiliza um vetor de listas encadeadas (buckets) para lidar com colisões. Cada bucket armazena uma lista de entradas, onde cada entrada é um par chave-valor.
- **Inserção:** Para inserir uma nova entrada, primeiro é calculado um índice para a entrada usando uma função hash. Esta função transforma a chave (neste caso, a palavra) em um número inteiro que indica a posição na tabela hash. Se a posição calculada já estiver ocupada por outra entrada (uma colisão), a nova entrada é adicionada na mesma posição, mas em uma lista encadeada. Isso significa que cada posição na tabela hash pode conter uma lista de entradas.A nova entrada é armazenada na tabela hash, seja diretamente na posição calculada ou na lista encadeada naquela posição.
- **Busca na Estrutura Hash:** Para buscar uma entrada, calcula-se o índice usando a função hash aplicada à chave (palavra) que se deseja encontrar.A posição calculada na tabela hash é verificada. Se houver uma lista de entradas (devido a colisões), a busca percorre essa lista procurando pela entrada com a chave correspondente. Se a entrada for encontrada, ela é retornada; caso contrário, a busca continua até o final da lista na posição calculada ou até que seja confirmado que a entrada não está presente.

3.1.3. Implementação dos contadores:

Os contadores foram implementados para medir e monitorar o desempenho das operações de inserção e busca em duas estruturas de dados: a árvore Patricia e a Tabela Hash. Essas métricas foram cruciais para fazer a avaliação do desempenho relativo de cada estrutura de dados para diferentes operações, conforme solicitado na especificação do trabalho prático.

a) Número de Comparações de Inserção:

Objetivo: Avaliar a eficiência de cada estrutura de dados durante a inserção de novos termos.

Árvore Patricia:

- **Inserção:** Durante a inserção de um novo termo na árvore Patricia, a estrutura é navegada até o ponto de inserção ou até encontrar um nó existente onde a inserção deve ocorrer. O número de comparações mede quantas vezes a árvore foi percorrida até encontrar o local correto para o novo termo.

Tabela Hash:

- **Inserção:** Durante a inserção em uma tabela hash, o termo é mapeado para um índice na tabela usando uma função hash. Se houver colisões (vários termos mapeados para o mesmo índice), a tabela pode precisar lidar com essas colisões, assim, aumentando o número de comparações.

b) Número de Comparações de Consulta:

Objetivo: Avaliar a eficiência de cada estrutura de dados durante a consulta de termos de busca.

Árvore Patricia:

- **Consulta:** Durante a consulta, a árvore Patricia é percorrida para encontrar o termo. O número de comparações mostra quantas vezes a árvore foi percorrida para localizar o termo desejado.

Tabela Hash:

- **Consulta:** Durante a consulta, a função hash é usada para calcular o índice do termo, e o termo é procurado nesse índice. Se o termo não estiver presente, o número de comparações pode ser maior devido à necessidade de verificar as entradas de colisão.

4. Considerações finais

Durante o desenvolvimento do projeto, foram observadas algumas dificuldades e facilidades de implementação em ambas as estruturas. Nossa principal dificuldade foi implementar e identificar onde estavam os erros para atualizar as ocorrências dos ingredientes ao longo dos arquivos nas duas estruturas. Tanto é que, na estrutura Patrícia, a saída da criação dos índices invertidos não saiu como esperado. Encontramos um certo grau de dificuldade também para implementar a árvore Patricia devido à sua estrutura detalhada. Por outro lado, já possuíamos facilidade em lidar com a compilação por meio do Makefile e com a implementação de TADs, o que economizou um pouco de tempo no desenvolvimento do projeto. Portanto, conseguimos entregar tudo o que foi pedido na especificação do trabalho e no tempo dado, mesmo com todas as dificuldades encontradas.

5. Referências Bibliográficas:

1. MANNING, C.D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. Cambridge University Press, 2008. Disponível em: Google Books. Acesso em: 4 ago. 2024..
2. BÜTTCHER, S.; CLARKE, C.L.A.; CORMACK, G.V. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, 2016. Disponível em: Google Books. Acesso em:4 ago. 2024.
3. ZIVIANI, N. *Implementação de Estruturas de Dados: Árvores PATRICIA*. 2021. Disponível em: Implementação PATRICIA - BYZiviani. Acesso em: 4 ago. 2024.
4. YOUTUBE. *Introdução a Árvores PATRICIA*. Disponível em: Introdução a Árvores PATRICIA - YouTube. Acesso em: 4 ago. 2024
5. BUENO, M. *Estruturas de Dados - Árvores Patricia*. Universidade Federal de Viçosa. Disponível em: Marcio Bueno - Estruturas de Dados. Acesso em: 4 ago. 2024
6. CARMEM, C. *Estruturas de Dados - Árvores Patricia*. Disponível em: Universidade Federal do Paraná. Acesso em: 4 ago. 2024.
7. SILVA, Glaucia Braga e. *Funções de transformação para hash*. Disponível em: https://ava.ufv.br/pluginfile.php/783178/mod_folder/content/0/02-Fun%C3%A7%C3%B5es%20de%20Transforma%C3%A7%C3%A3o%20para%20Hash.pdf?forcedownload=1. Acesso em: 4 ago. 2024.
8. SILVA, Glaucia Braga e. *Tabela hash*. Disponível em: https://ava.ufv.br/pluginfile.php/783178/mod_folder/content/0/01-Tabela%20Hash.pdf?forcedownload=1. Acesso em: 4 ago. 2024.
9. SILVA, Glaucia Braga e. *Tratamento de colisões em hash*. Disponível em: https://ava.ufv.br/pluginfile.php/783178/mod_folder/content/0/03-Tratamento%20de%20Colis%C3%B5es%20em%20Hash.pdf?forcedownload=1. Acesso em: 4 ago. 2024.