PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Bacharelado em Ciência da Computação

Luiz Fernando Antunes da S. Frassi

Trabalho de Aeds: TP1->TP2->TP3

Belo Horizonte

2023

Luiz Fernando Antunes da S. Frassi

Trabalho de Aeds: TP1->TP2->TP3

Compilado explicando como os Trabalhos Práticos foram desenvolvidos.

Belo Horizonte

2023

RESUMO

Neste trabalho foram feitas as tres etapas solicitadas: Etapa 1: Criação da base de dados, Ordenação Externa, Hash e Lista Invertida; Etapa 2: Compactação com Huffman e LZW; Etapa 3: Casamento de Padrões(KMP e Boyer-More) e Criptografia (RSA);

Palavras-chave:Casamento de Padrões,Compactação, CRUD, .

SUMÁRIO

1 TP1

Para este TP, foi utilizada a base de dados 'airplane-crash', contendo 4966 registros de 1908 até 2019.

1.1 Crude

O **CRUD** foi criado utilizando o seguinte esquema para armazenamento sequencial no banco de dados: ID do registro, Data, Hora, Rota, Registro de voo, Total de embarques, passageiros embarcados, tripulação, Total de fatalidades, fatalidades de passageiros, fatalidades na tripulação, e LOA referentes à localização, operação e tipo da aeronave. Todos os registros possuem um marcador de lápide e um valor com o tamanho do registro na frente. Métodos do **CRUD** criados: para inserção de registro, para encontrar um registro específico, para mostrar todo o banco, para apagar um registro ou para atualizar um registro e a função que puxa todo o csv padrão para o banco de dados(caso o banco esteja vazio).

1.2 ORDENAÇÃO EXTERNA e INDEXAÇÃO

Foi realizada a Intercalação Balanceada Comum, onde os registros são intercalados em dois arrays de dados, ordenando de 4 em 4. A intercalação é refeita mais 2 vezes e juntada no final com os registros ordenados. Além disso, foi implementada a Intercalação Balanceada com Blocos de Tamanho Variável. Toda a ordenação ou atualização com CRUD é realizada no banco de dados em uma tabela hash.Podendo tabem fazer pesquisa de id pela tabela hash.

1.3 Lista invertida

Foi criada a lista invertida que, por meio do LOA (Location, Operator e AcType), encontra o registro no banco de dados. Isso difere da busca padrão, que é realizada através do ID.

2 TP2

Compressão de dados e casamento de padrão.

2.1 Compressão

Foram aplicados o algoritmo de Huffman, montando sua árvore de acordo com o peso de cada caractere (quantidade de vezes repetidas), onde o caminho de cada caractere na árvore forma uma sequência de 0 e 1 que será substituída no arquivo. Além disso, foi utilizado o algoritmo LZW, com o dicionário gerado a partir do próprio texto. Todas as compactações são salvas em "cmp"e a descompactação em "dcmp".

2.2 Casamento de padrão

No contexto de casamento de padrão, foram aplicados os algoritmos KMP, no qual é primeiro calculada a função de falha e, em seguida, a busca é realizada conforme o padrão. E Boyer-Moore, que verifica a chave do fim ao início ao longo do texto, e seu método de pesquisa na chave indica quantos caracteres devem ser pulados.

3 TP3

Entre dois algoritmos simples e um complexo, foi escolhido o RSA, onde o texto criptografado é guardado em uma pasta chamada 'cript' e da mesma forma, a descriptografia também é guardada nesta pasta.

3.1 Criptografia em RSA

Para a criptografia com RSA, primeiramente são definidos valores padrão para as variáveis 'p' e 'q', sendo dois números primos de tamanho significativo. Em seguida, para 'n', é calculado o produto de 'p' e 'q', enquanto para 'z', é feito o produto de '(p-1)(q-1)'. A variável 'd' representa o primo em relação a 'z'; no caso, a função 'primoEmRelacao' obtém o maior número primo em relação a 'z'. Por fim, 'e' deve ser um valor que satisfaça a seguinte equação: (e*d) Mod z == 1. Nesse contexto, a função 'functionE' retorna um valor que satisfaça a equação. Por último, o valor ASCII de cada caractere é elevado pelo valor 'e' e é retirado o módulo de 'n'; esse valor resultante é a criptografia do caractere original.

Na descriptografia, os mesmos valores de 'p' e 'q' são gerados para criar chaves correspondentes. Sobre o valor criptografado, é realizada a mesma operação, mas desta vez o valor é elevado por 'd', e em seguida, é retirado o módulo de 'n'. Essa operação resulta no valor ASCII do caractere original.

4 TESTES E RESULTADOS

4.1 Tp1

Na etapa 1, apesar da demora na resposta dos métodos de ordenação, principalmente devido à atualização na tabela hash, eles desempenham bem sua função de ordenação. O CRUD opera normalmente, inserindo novos registros sempre ao final do banco, reescrevendo valores sobre o original (caso sejam iguais ou menores) e utilizando o sistema de lapidação para apagar itens. Funções como a de ordenação têm permissão para destruir arquivos marcados com lapidação, eliminando assim lacunas no banco de dados.

4.2 Tp2

Na etapa 2, foram implementados os códigos de Huffman e LZW, sendo que o Huffman apresentou uma perda de espaço devido à diversidade de caracteres no CSV, tornando o caminho da árvore de Huffman mais longo e menos eficiente. Por outro lado, o LZW proporcionou um ganho de quase 50

4.3 Tp3

Na etapa 3, o algoritmo RSA demonstra ser mais extenso que o original, atribuindo, na maioria das vezes, um número de quatro dígitos a cada caractere com a configuração atual de 'p' e 'q'. Essa ampliação pode ser justificada pelas características do algoritmo RSA e pelas configurações específicas escolhidas para 'p' e 'q'.

5 CONCLUSÃO

Todos os requisitos foram cumpridos, e seus algoritmos foram implementados, sendo apenas o algoritmo de Huffman que não obteve bons resultados.

Acesso ao código: "https://github.com/Luix-F/Tp3.git".