Universidade Federal da Bahia
MATA54 - Estruturas de Dados e Algoritmos II
Quarto Trabalho Prático
Prof. Flávio Assis
Semestre 2021.2 - 6 de novembro de 2021

Mecanismo de Busca de Palavras em Arquivos

1 Descrição Geral do Trabalho

Neste trabalho será implementado um programa de busca de palavras em arquivos. De uma forma simplificada, o programa implementará a função encontrada em sistemas de busca em páginas *web* (representadas, neste trabalho, como arquivos).

O usuário irá informar um conjunto de arquivos de texto. O programa manterá um índice, contendo todas as palavras que ocorrem nos arquivos. Esse índice será usado para se obter uma lista dos arquivos em que uma determinada palavra ocorre. Adicionalmente, o programa informará em que posição do arquivo as palavras ocorrem.

O índice deverá ser implementado como uma *trie*. Essa *trie* manterá, para cada palavra, uma lista de até três nomes de arquivos em que a palavra ocorre. Para se encontrar em que posições de um arquivo uma determinada palavra ocorre, deverá ser usado o algoritmo KMP (*Knuth-Morris-Pratt*).

2 Formato da Entrada e Saída

O programa deve ler as operações da entrada padrão e escrever na saída padrão. Somente letras ocorrerão em palavras, sem acento e sem cedilha. Todos os caracteres na entrada serão minúsculos. Uma palavra poderá conter, no máximo, 30 caracteres. Todas as saídas geradas devem conter apenas letras minúsculas.

A entrada consistirá de uma sequência de operações. O conjunto de operações a serem implementadas e seus formatos são os seguintes:

1. **inserção de arquivos na base:** esta operação conterá inicialmente uma linha contendo a letra i, seguida de outra linha contendo o nome de um arquivo (sequência de letras e, possivelmente, um ponto, de tamanho máximo 20).

   O arquivo fornecido será um arquivo texto. O programa deve inserir todas as palavras do arquivo na *trie*. Para cada palavra, é mantida uma lista de até três arquivos em que a palavra ocorre. Os arquivos mantidos nessa lista são os três primeiros arquivos informados que contêm a palavra. Se houver mais de três arquivos, os demais serão ignorados.

   Caso o arquivo indicado na operação não exista, o programa deve apresentar na saída a sequência *'arquivo nao encontrado:'*, seguida de um espaço, seguido do nome do arquivo. Se o arquivo existir, o programa deve indicar na saída a sequência de caracteres *'arquivo processado com sucesso:'*, seguida de um espaço, seguido do nome do arquivo.

2. **consulta de palavra:** esta operação conterá inicialmente uma linha contendo a letra c, seguida de uma outra linha, que conterá uma palavra.

   Essa operação irá verificar se a palavra ocorre na base. Se ocorrer, o programa deve gerar na saída a sequência de caracteres *'ocorrencias da palavra:'*, seguida de um espaço, seguido da palavra, seguida do caractere *':'* (dois pontos).

   Em seguida, para cada arquivo em que a palavra ocorrer, o programa deverá gerar a seguinte saída. Inicialmente gerará a sequência de caracteres *'arquivo:'*, seguida de um espaço, seguido

do nome do arquivo. Em seguida, as ocorrências da palavra devem ser apresentadas, uma em cada linha, da seguinte forma: sequência de caracteres *'linha:'*, seguida de um espaço, seguido do número da linha em que a palavra ocorre, seguido de um espaço, seguido da sequência de caracteres *'posicao:'*, seguida de um espaço, seguido da posição na linha em que a palavra ocorre. A primeira linha do arquivo deve ser considerada como sendo a linha 1. A posição na linha indica qual é a posição do primeiro caractere da palavra na linha. Considere que o primeiro caractere da linha está na posição 1.

A saída desta operação deve considerar os arquivos na ordem em que foram inseridos no sistema. As ocorrências das palavras devem ser apresentadas seguindo a ordem do texto.

A busca pelas ocorrências das palavras deve ser feita utilizando o algoritmo KMP. Uma palavra somente ocorrererá de forma completa em uma única linha (ou seja, uma palavra não irá começar em uma linha e terminar em outra).

Se a palavra não ocorrer em nenhum arquivo, o programa deve gerar a sequência *'palavra nao ocorre na base:'*, seguida de um espaço, seguido da palavra.

3. **arquivos em que uma palavra ocorre:** esta operação conterá inicialmente uma linha contendo a letra a, seguida de outra linha contendo uma palavra.

   Se a palavra existir na *trie*, o programa deve gerar na saída a sequência de caracteres *'palavra:'*, seguida de um espaço, seguido da palavra. Na linha seguinte, deve gerar a sequência *'arquivos em que ocorre:'*, seguida de um espaço, seguido dos nomes dos arquivos em que a palavra ocorre, na mesma linha, separados por um espaço. Não deve haver espaço após o nome do último arquivo. Os nomes dos arquivos deve seguir a ordem em que foram inseridos pelo usuário.

   Se a palavra não existir na *trie*, o programa deve gerar a sequência *'nao ha ocorrencia da palavra:'*, seguida de um espaço, seguido da palavra.

4. **tabela pi:** esta operação conterá inicialmente uma linha contendo a letra p, seguida de uma outra linha, contendo uma palavra.

   Esta operação deve apresentar, na saída, a tabela *pi* (π), **como apresentada no livro de Cormen et al. e nas aulas**. O programa deve gerar, na saída, inicialmente a sequência de caracteres *'tabela pi para a palavra:'*, seguida de um espaço, seguido da palavra, seguida de dois pontos (':'). A seguir, o programa deve apresentar o valor da tabela em si, da seguinte forma: para cada letra da palavra, uma em cada linha, a partir da primeira letra, inicialmente a letra entre aspas simples, dois pontos (':'), um espaço, e o valor da tabela pi para aquela letra.

5. **número de nós da *trie*:** esta operação consiste apenas de uma linha, contendo a letra n.

   Esta operação deve apresentar o número de nós da *trie*. Nesta contagem, deve-se incluir os nós correspondentes às marcas de fim de palavra.

   Esta operação gera na saída a sequência de caracteres *'numero de nos na trie:'*, seguida de um espaço, seguido do número de nós da *trie*. Se a *trie* estiver vazia, o número de nós a ser apresentado é zero.

6. **término da sequência de comandos:** a sequência de comandos será terminada por uma linha com a letra 'e'.

3 Procedimentos Gerais

3.1 Persistência

O estado dos dados do programa deve ser mantido persistente. Ou seja, inicialmente o programa deve criar os arquivos necessários para manter os dados persistentes. A partir de então, quaisquer alterações nos dados devem ser refletidas no conteúdo dos arquivos. O estado dos dados deve ser mantido de uma

invocação a outra do programa. Quando uma nova sequência de testes do programa for ser criada, os arquivos com os dados serão apagados.

Todos os arquivos utilizados pelo programa para armazenar os dados deverão ter extensão *".dat"*.

3.2 Considerações Gerais Adicionais

Como no caso dos trabalhos anteriores desta disciplina, **não se pode considerar que toda a base de dados cabe na memória principal**. Especificamente, não se pode considerar que a *trie* caiba, por completo, na memória principal. Trabalhos que não atenderem este requisito serão considerados inaceitáveis.

4 Observações

Trabalho individual ou em dupla. O trabalho deve ser entregue através da plataforma Moodle, em um único arquivo. Este arquivo deve conter:

1. os arquivos com a implementação (apenas código fonte). Programas sem comentários (realizados de forma adequada) terão sua nota reduzida.

2. um texto, em formato pdf, descrevendo de que forma a *trie* foi mantida persistente e como a limitação de memória foi atendida (ou seja, como a *trie* é mantida de forma parcial na memória principal).

Importante:: Caso o trabalho seja feito em dupla, o nome dos participantes deve estar claro nos arquivos entregues. Não será permitido acrescentar nome de aluno(a) após a entrega do trabalho!

Data de entrega: 29/11/2021. Não será possível adiar a entrega do trabalho além desta data!

Linguagens de programação permitidas: C, C++, Java ou Python.

Observação Importante: Para as linguagens C, C++ e Java, somente trabalhos feitos utilizando os seguintes compiladores serão aceitos:

- C: gcc ou djgpp
- C++: g++ ou djgpp
- Java: compilador java do JDK (mais recente)

Não serão compilados trabalhos em outros compiladores! Erros ocasionados por uso de diferentes compiladores serão considerados erros do trabalho!

No caso de Python, o aluno deve indicar a versão utilizada.