Universidade Federal da Bahia
MATA54 - Estruturas de Dados e Algoritmos II
Quarto Trabalho Prático
Prof. Flávio Assis
Semestre 2022.1 - 9 de maio de 2022

Árvore k-d e Busca de Padrões

1 Descrição Geral do Trabalho

Neste trabalho deverá ser implementada uma extensão ao terceiro trabalho prático. Nos registros da implementação realizada no terceiro trabalho há um campo para armazenamento do nome de um arquivo. Neste quarto trabalho, estes arquivos existirão. Cada um deles conterá um texto. O objetivo do trabalho é permitindo buscas que envolvam a ocorrência de palavras nos textos. A estrutura definida para o terceiro trabalho prático deverá ser mantida (índice implementado como árvore k-d), apenas complementada com as funcionalidades para busca de padrões em texto.

2 Formato de Entrada e Saída

O formato para entrada e saída definida para o terceiro trabalho deverá ser mantido neste trabalho. Ou seja, todas as convenções de entrada (por exemplo, para a criação da árvore k-d) e todas as operações definidas devem estar implementadas.

Adicionalmente, as operações descritas abaixo devem ser implementadas. Como nos trabalhos anteriores, os formatos descritos descrevem como as entradas, de fato, serão. Não há necessidade de validar se seguem o formato indicado.

Nomes de autores e títulos de livros devem seguir o mesmo formato definido para o terceiro trabalho. Quando ocorrer uma palavra como parâmetro de uma operação, essa palavra será formada apenas por caracteres minúsculos, sem acento e sem cedilha, e terá, no máximo, trinta caracteres. Os textos das obras estarão armazenados em arquivos do tipo *texto*. Nesses arquivos haverá apenas letras minúsculas, sem acento e sem cedilha. Pode ocorrer, no entanto, pontuação.

1. **arquivos (obras) em que uma palavra ocorre:** esta operação conterá inicialmente uma linha contendo a letra f, seguida de outra linha contendo uma palavra.

   Essa função retorna, para cada arquivo que contiver a palavra informada, o nome do autor e o título do livro correspondente.

   Para cada arquivo em que a palavra ocorre, 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 *'titulo:'*, seguida de um espaço, seguido do título da obra correspondente ao arquivo. Na linha seguinte, deve gerar a sequência *'autor:'*, seguida de um espaço, seguido do nome do autor da obra. Na linha seguinte, a sequência *'ano:'*, seguida de um espaço, seguido do ano da obra.

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

2. **consulta de palavra:** esta operação conterá quatro linhas. A primeira linha conterá a letra w. A segunda linha conterá uma palavra. A terceira linha conterá o nome de um autor. A quarta linha conterá o título de uma obra.

   Se não houver obra com o título e nome de autor indicados, o programa deve apresentar na saída a sequência de caracteres *'obra inexistente: titulo:'*, seguida de um espaço, seguido do título da

obra, seguido de um espaço, seguido da sequência *'- autor:*, seguida de um espaço, seguido do nome do autor indicado.

Caso haja a obra, a operação deverá apresentar na saída em uma linha a sequência de caracteres *'ocorrencia(s) da palavra:'*, seguida de um espaço, seguido da palavra, seguida de dois pontos *':'* (dois pontos). Em seguida, as ocorrências da palavra, se houver, 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.

Está-se considerando que os campos referentes ao autor e ao título de uma obra, tomados em conjunto, formam uma chave primária. Portanto, não haverá mais do que uma ocorrência de obra de um mesmo autor com um mesmo título.

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).

3. **tabela pi:** esta operação conterá inicialmente uma linha contendo a letra m, 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.

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

3 Observações

- O programa deve manter as atualizações em arquivo. A correção levará em consideração que o estado dos dados é persistente. Com isto, um teste pode ser feito, por exemplo, inserindo-se um registro, terminando a execução do programa e fazendo uma consulta ao registro em nova invocação do programa. Neste caso o registro deve ainda estar no arquivo.

- Lembre-se de que é assumido que a memória principal é insuficiente para armazenar todos os dados. Nesse trabalho, assume-se que o índice da árvore *k-d* cabe na memória principal, *mas não o conteúdo das páginas*. Portanto, por exemplo, uma implementação que mantém uma estrutuda com todos os dados em memória principal e a salva por completo no arquivo será considerada inaceitável.

- Os arquivos devem ser armazenados em formato binário.

- O programa não pode gerar nenhum caractere a mais na saída, além dos indicados acima. Em particular, o programa não deve conter menus.

- Não pode haver espaço entre linhas na saída. A saída deve apresentar os caracteres em letras minúsculas.

- Todos os arquivos criados pelo programa dever ter extensão ".dat".

- Trabalho individual ou em dupla.

- Além do código fonte do programa, deve ser entregue também um relatório sucinto, indicando a lógica de criação da árvore k-d e como ela foi armazenada em memória secundária.

  Importante: Se não houver entrega desse relatório, a nota do trabalho pode ser **bastante** afetada, uma vez que o relatório é necessário para se entender como a implementação foi feita.

- Data de entrega: **26/06/2022 (prazo firme, em função do fim do semestre)**

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

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

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

  No caso de Python, deve ser indicada a versão utilizada.

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

- O trabalho deve ser entregue através do *moodle*.