Universidade Federal de Mato Grosso do Sul
Facom - Faculdade de Computação

Estruturas de Dados – 2021/1
Profa. Bianca de Almeida Dantas

Trabalho – Índice Invertido

1 Descrição

A construção de índices invertidos para documentos é uma parte importante do processo realizado por diferentes ferramentas de busca, tais como Google e Yahoo!. Considerando uma coleção de documentos, um *índice invertido* é uma estrutura que contém uma entrada para cada termo encontrado em algum dos documentos. A estrutura associa, para cada termo, pares com o formato `<contador, docId>`, onde `docId` é o identificador do documento e `contador` é o número de vezes que esse termo aparece no documento em questão. Quando um documento não contém um termo, ele não precisa aparecer nos pares referentes a tal termo.

1.1 Formato da Entrada e da Saída

A entrada do programa consiste em um arquivo texto contendo, na primeira linha, o número de arquivos da coleção e os nomes desses arquivos nas linhas subsequentes. O formato desse arquivo deve ser como segue:

```
N
arquivo1.txt
arquivo2.txt
...
arquivoN.txt
```

Assuma que os arquivos, caso existam, estão no diretório corrente de execução. O seu programa deve processar os arquivos e construir o índice invertido e, para isso, precisa atribuir uma identificação numérica única, o `docId`, para cada arquivo, na ordem em que aparece no arquivo de entrada. A saída consiste em um arquivo contendo uma linha para cada termo encontrado nos arquivos; cada linha deve conter o próprio termo e o par ordenado obtido pelo algoritmo de obtenção do índice, separados por um espaço. Ao final da linha correspondente ao último termo deve-se inserir uma nova linha.

Para exemplificar, suponha que o conteúdo do arquivo de entrada é:

```
2
texto1.txt
texto2.txt
```

Os dois arquivos estão presentes no diretório de execução e seus conteúdos são os seguintes:

texto1.txt:
Quem casa quer casa. Porem ninguem casa. Ninguem quer casa tambem. Quer apartamento.

texto2.txt:
Ninguem em casa. Todos sairam. Todos. Quer entrar? Quem? Quem?

Atribuindo os identificadores 1 e 2 aos dois textos, respectivamente, o arquivo de saída seria o seguinte:
apartamento 1 1
casa 4 1 1 2
em 1 2
entrar 1 2
ninguem 2 1 1 2
porem 1 1
quem 1 1 2 2
quer 3 1 1 2
sairam 1 2
tambem 1 1
todos 2 2

Observe que:

- O arquivo de saída deve ser **exatamente** como descrito, nenhuma informação a mais ou a menos.
- Os termos devem aparecer no arquivo de saída em ordem alfabética;
- Uma palavra é considerada como uma sequência de letras;
- Sinais de pontuação devem ser ignorados;
- Os textos não serão acentuados;
- Palavras com letras maiúsculas devem ser primeiramente transformadas para minúsculas antes da inserção no índice.
- Caso algum dos arquivos não exista, ele deve ser ignorado na atribuição do código de identificação. Os códigos devem ser atribuídos a partir de 1 apenas para arquivos válidos.

Para armazenamento do índice invertido, deverá ser usada a estrutura de uma **B-Árvore**, cuja implementação para chaves inteiras foi fornecida na página EAD da disciplina (no tópico 4). O seu programa será compilado utilizando o compilador `g++` do Unix como segue:

```
g++ -Wall *.cpp
```

e a execução do programa será realizada por linha de comando passando como parâmetro o arquivo de entrada, como segue:

```
./exec_trabalho arq_entrada.txt,
```

onde `exec_trabalho` é o nome de seu executável e `arq_entrada.txt` é o nome do arquivo de entrada. O arquivo de saída deverá ser gravado no mesmo diretório com o mesmo nome acrescido de `.out` ao final. No caso do exemplo anterior, deve ser gerado um arquivo com o nome `arq_entrada.txt.out`.

2 Grupos e Entrega

O trabalho pode ser feito em grupos de, no máximo, 3 estudantes e a entrega deve ser realizada até o dia **25/06/2021**, em um arquivo compactado, através de postagem na página do AVA da disciplina.

Devem ser entregues os códigos fonte implementados e um arquivo *readme* contendo os nomes dos integrantes do grupo, todos compactados em um único arquivo (.zip ou .rar).

3 Avaliação

- Programas que não compilarem ou não gerarem os arquivos de resultado não serão considerados e receberão nota zero.
- Os programas serão executados com N arquivos de entrada. Para cada saída correta o trabalho receberá a pontuação de $10/N$.
- A correção será realizada comparando o arquivo de saída gerado pelo seu programa com um arquivo com a saída esperada. **Somente a igualdade será considerada válida**.
- **Trabalhos copiados ou fortemente inspirados em outros receberão nota zero.**