UNIVERSIDADE FEDERAL DE MINAS GERAIS
PROGRAMAÇÃO E DESENVOLVIMENTO DE SOFTWARE II

Trabalho Prático: Máquina de Busca

Thiago Ferreira de Noronha

4 de Outubro de 2018

A base para o funcionamento de uma máquina de busca, tais como Google e Yahoo!, é a construção de um índice invertido para uma coleção de documentos. Dada a coleção de documentos, um índice invertido é uma estrutura contendo uma entrada para cada palavra (termo) que aparece em pelo menos um documento. Para cada palavra, o índice invertido armazena a lista de documentos que a contém. Por exemplo, dados os seguintes documentos:

Documento "d1.txt":

```
Quem casa quer casa.  Porem ninguem casa.
Ninguem quer casa tambem.  Quer apartamento.
```

Documento "d2.txt":

```
Ninguem em casa.  Todos sairam.  Todos.
Quer entrar?  Quem?  Quem?
```

Documento "d3.txt":

```
Quem esta no apartamento?  Ninguem, todos sairam.
```

O índice invertido seria:

apartamento	d1.txt	d3.txt	
casa	d1.txt	d2.txt	
em	d2.txt		
entrar	d2.txt		
esta	d3.txt		
ninguem	d1.txt	d2.txt	d3.txt
no	d3.txt		
porem	d1.txt		
quem	d1.txt	d2.txt	d3.txt
quer	d1.txt	d2.txt	
sairam	d2.txt	d3.txt	
tambem	d1.txt		
todos	d2.txt	d3.txt	

Neste trabalho você deve projetar um sistema que implementa uma máquina de busca utilizando um índice invertido. O trabalho consiste nas 3 partes, descritas a seguir:

Parte 1: Leitura de arquivos (1 ponto)

O sistema recebe como entrada um conjunto de arquivos de texto, que devem ser lidos, palavra após palavra, para construir o índice invertido. Detalhes:

- Você pode assumir que os arquivos contém apenas caracteres que são letras (a-z e A- Z), números (0-9), e caracteres de pontução (",", ".", "?", etc.).
- Você pode assumir que o texto nao tem acentos nem "ç".
- Apos ler cada palavra, você deve (i) transformar todas as letras maiúsculas em minúsculas e (ii) apagar todos os caracteres que não são letras ou números. Por exemplo, depois de ler "Guarda-Chuva?", você deve transformá-la em "guardachuva", antes de inseri-la no índice invertido. Desta forma, a mesma palavra apresentada com letras minúsculas ou maiúsculas, ou que estão adjacentes a pontuação, não serão diferenciadas.

Parte 2: Estruturas de dados do índice invertido (5 pontos)

Para implementar o índice invertido você deve utilizar uma estrutura de dados do tipo dicionário (map). Dicionários são estruturas associativas que armazenam elementos combinando uma chave com um valor. A chave é usada exclusivamente para identificar o elemento, que contém um valor mapeado. Podemos fazer uma analogia com a lista telefônica, em que o nome da pessoa é a chave, e o telefone da pessoa é o valor associado ao nome. Neste trabalho, as chaves são todas as palavras contidas nos documentos e o valor associado a cada chave é o conjunto (set) com os nomes dos documentos.

Parte 3: Consultas simples (4 pontos)

Nesta parte do trabalho você utilizará o índice invertido construído para realizar consultas de UMA palavra. Dada uma palavra, o seu buscador deve imprimir na tela a lista de documentos que contém a palavra em ordem alfabética.
Por exemplo: para a consulta por `apartamento` o seu programa deve imprimir:

```
d1.txt
d3.txt
```

Parte 4: Documentação e Entrega (10 pontos)

A documentacao nao devera exceder o limite de 10 paginas, sendo submetida no formato PDF juntamente com o codigo fonte via GitHub. Cada aluno deve criar seu próprio repositório público no site e enviar o link para o meu email (mvaguimaraes@gmail.com). Alem disso, a documentacao devera contemplar o seguinte requisito:

• Introducao com uma explicacao clara e objetiva de como o problema que foi resolvido, justificando os algoritmos e as estruturas de dados utilizadas. Para auxiliar nessa atividade utilize pseudocodigos, diagramas e demais figuras que achar conveniente. Nao e necessario incluir trechos de codigo da sua implementacao e nem mostrar maiores detalhes da sua implementacao, exceto quando esses influenciam no seu algoritmo principal.

Parte 5: Testes de unidade (10 pontos)

Desenvolvam testes unitários para os métodos que forem criados seguindo as instruções dadas em sala. Tentem cobrir todos os casos que vocês imaginarem. A taxa de cobertura mínima de código deve ser de 75%.