# Universidade Federal de Goiás
# Bacharelado em Ciência da Computação
# Tópicos - II 2011-2

## Trabalho de Programação
### Implementação de uma Máquina de Busca

prof. Thierson Couto Rosa

## Objetivos

Este trabalho corresponde à implementação dos módulos de indexação e processamento de consultas de uma máquina de busca. O objetivo principal é o de dar a oportunidade aos alunos de experimentarem técnicas de construção de índice invertido e de processamento de consultas, estudadas durante o curso, e de proporem novas técnicas para implementação desses dois módulos de uma máquina de busca.

## Construção do índice

O módulo de construção do índice deve receber como entrada um conjunto de quádruplas $\langle term, p, pos, docId \rangle$, onde $term$ é um termo encontrado em um documento, $p$ é a parte do documento onde o termo ocorre (T - título do documento, C - corpo do documento), $pos$ é a posição onde o termo ocorre na parte correspondente e $docId$ é um número inteiro que corresponde a um identificador único do documento onde $term$ ocorre.

A Tabela 1 ilustra um exemplo de entrada para o módulo de construção do índice. A primeira quádrupla corresponde à ocorrência do termo *Use* na primeira posição do título do documento 2. A quinta quádrupla corresponde à ocorrência do termo *This* na primeira posição do corpo do documento 2.

O módulo de construção do índice invertido deve ser implementado de tal modo que as células das listas invertidas (*postings*) contenham informações necessárias para pelo menos os seguintes tipos de processamentos de consultas:

a) Processamento de consultas com conectivos lógicos (OU, E, Não).

b) Processamento de ordenação pelo método dos cossenos.

c) Processamento por frases.

| |
|---|
| Use T 1 2 |
| Made T 2 2 |
| of T 3 2 |
| Technical T 4 2 |
| Libraries T 5 2 |
| This C 1 2 |
| report C 2 2 |
| .... |

Tabela 1: Exemplo de entrada para o módulo de indexação.

# Processamento de Consultas

O módulo de processamento de consultas deverá processar consultas por conjunto de termos, consultas com conectivos lógicos e consultas por frase. As consultas formadas por conjunto de termos, sem conectivos entre eles, devem ser processadas utilizando-se o método dos cossenos. As consultas com conetivos lógicos devem ser tratadas como consultas lógicas. Finalmente, uma consulta formada por um conjunto de termos entre aspas deve ser tratada como uma consulta por frase.

A saída do processamento de uma consulta deve ser um arquivo com codificação ASCII, contento um identificador numérico de documento ($docId$) por linha. No caso de consultas por conjunto de termos, a ordenação dos números de documentos a serem apresentados pelo usuário deve ser em ordem decrescente do valor dos cossenos entre os ângulo entre os vetores correspondentes ao documento e a consulta.

O processador de consultas deve ser testado utilizando-se todas as consultas disponíveis em cada coleção de teste. Os arquivos com as respostas devem ser armazenados em um mesmo diretório para que possam ser avaliados por programas de avaliação de resultados de máquinas de busca, a serem entregues futuramente pelo professor. Estes programas geram valores de medidas de precisão e revocação e gráficos de precisão e revocação.

# Coleções de teste

Neste trabalho serão utilizadas cinco coleções de teste:

- Library and Information Science Abstracts - LISA
- Medline collection - MED.
- Comité Interministériel pour la Société de l'Information - CISI
- NPL
- Coleção de trechos de artigo da revista Time -TIME

As coleções estão compactadas no arquivo colecoes.tar.gz e pode ser copiado através do endereço: www.inf.ufg.br/ ∼thierson/. Essas coleções estão disponíveis em diretórios. Cada diretório contém os seguintes arquivos e subdiretórios:

- entradaIndexador.txt - arquivo contendo as quádruplas de entrada para o módulo de indexação.
- consultas - diretório contendo um arquivo por consulta. O nome de cada arquivo é formado pelo número da consulta e a extensão ".qurey".
- relevantes - diretório contendo um arquivo correspondente a cada consulta. Cada arquivo é formado pelo número da consulta e a extensão ".query". Um arquivo desse diretório contém em cada linha um *docId* de um documento considerado relevante para a consulta correspondente.
- parsers - diretórios com programas para gerar o arquivo entradaIndexador.txt, o diretório de consultas e o diretório de documentos relevantes.

Os documentos das coleções CISI e LISA possuem título e corpo. As demais não possuem título. As consultas em todas as coleções, exceto na coleção CISI, são formadas por conjunto de termos.

O aluno deve executar as consultas como estão disponíveis, para avaliar o processamento utilizando o método dos cossenos. Deverá escolher um conjunto de consulta de uma das coleções e modificá-lo, de modo a testar consultas por frase.

A coleção CISI disponibilizada contém apenas consultas lógicas. Para esta coleção não foi entregue o subarquivo com o conjunto de relevantes, pois para consultas lógicas devem ser retornados todos os documentos que satisfazem à consulta. As consultas lógicas da coleção CISI são formadas por expressões que podem são descritas pela gramática da Tabela 2.

| | | |
|---|---|---|
| *consulta* | → | *expr* ';' |
| *expr* | → | *operador* '(' *lista_expr* ')' \| |
| | | *termo* |
| *lista_expr* | → | *expr* ',' *expr le* |
| *le* | → | ',' *expr le* \| |
| | | $\epsilon$ |
| *operador* | → | '#' op |
| *op* | → | **and** \| **or** \| **not** |

Tabela 2: Gramática para expressões que compõem consultas lógicas na coleção CISI.

A seguir são apresentadas, como exemplo, duas consultas presentes da coleção CISI e que são aceitas pela gramática acima:

- #and (titles, #or (automatically, retrieving, problems, concerns, descriptive, approximate, difficulties, content, relevance, articles));

- #or (#and (group, mathematics), #and (athematics, #or (abstract, information, retrieval)));

O trabalho deve contemplar a implementação de um *parser* para estas consultas para o processamento das mesmas sobre a coleção CISI.

# Pontos extras

O trabalho podeerá receber até 3 pontos extras na avaliação, caso implemente pelo menos uma das funcionalidades extras descritas a seguir:

- Compactação e descompactação de listas invertidas – dois pontos.
- Proposta de uma nova heurística de ordenação de documentos utilizando combinação do método dos cossenos com outras informações, tais como: a parte do texto onde o termo ocorre, posição entre os termos da consulta, entre outras – três pontos.
- Otimização de consultas lógicas (reordenação dos operadores na expressão, para que o processamento seja mais rápido) – três pontos.

**Observação:** os pontos extras serão aplicados somente no caso em que as requisições especificadas nas seções *Construção do índice* e *Processamento de Consultas* forem atendidas.

# Penalidades

Será descontado um ponto por dia de atraso na entrega do trabalho. Além disso, o trabalho receberá nota zero em caso de cópias de parte do programa. Nesse caso a nota zero será aplicada a todos alunos envolvidos.