Universidade de Brasília
Departamento de Ciência da Computação

Projeto 4
Estruturas de Dados, Turma E, 1/2014
Prof. Díbio

Um índice invertido é um mapeamento de palavras até sua localização em um conjunto de documentos. A maioria das máquinas modernas de busca (e.g. *"google, altavista, bing"*) utilizam alguma forma de índice invertido para processar os pedidos de busca dos usuários. Na sua forma mais básica, um índice invertido é uma tabela de espalhamento (i.e. *"hashing"*) simples que relaciona palavras nos documentos a algum tipo de identificador do documento. Por exemplo, sejam três documentos:

Doc1:
Manga manga abacaxi pequi murici.

Doc2:
Manga doce.

Doc3:
pequi azedo.

Poder-se-ia construir o seguinte arquivo de índice invertido:
Manga -> Doc1, Doc2
manga -> Doc1
abacaxi -> Doc1
pequi -> Doc1, Doc3
murici -> Doc1
doce. -> Doc2
azedo. -> Doc3

No projeto 1 deste semestre (1/2014) seguimos os seguintes passos para calcular semelhança entre dois arquivos texto. Relembrando:
"...
Uma maneira possível para calcular a semelhança entre dois documentos seria:

1. ler arquivos dos documentos (doc1.txt, doc2.txt);
2. montar lista de palavras, e.g. ["a", "os", "um", ...];
3. calcular frequências das palavras, e.g. [["a", 212], ["os", 1200], ...];
4. ordenar a lista pelas frequências, e.g. [["os", 1200], ["a", 200], ...];
5. calcular o ângulo Θ entre os documentos.

..."

Um dos gargalos computacionais do método acima residia na função que calculava a frequência das palavras, pois se no limite todas as palavras fossem diferentes essa função seria de Ordem $O(n^2)$, buscando toda a lista para encontrar e contar frequências. Uma solução para melhorar a eficiência dessa

pesquisa seria com o uso de “dicionários”, ou funções de espalhamento (“*hashing*”) para fazer o mapeamento e a pesquisa em ordem linear, i.e. O(n).

Escreva um programa em linguagem C, o qual deverá ler dez (10) arquivos texto fornecidos (doc1.txt, doc2.txt, doc3.txt ... doc10.txt), gerar um arquivo (IndInvert.txt), o qual indicará como no exemplo as palavras (não repetidas) e os respectivos arquivos de localização. No programa deve haver uma função de espalhamento que lê o arquivo (IndInvert.txt) e gera os índices das palavras para os arquivos. Para este projeto os arquivos conterão somente caracteres alfanuméricos (maiúsculos e minúsculos), sem acentuação ou pontuação. Crie uma função de “*hashing*” que lide com as colisões pois o número máximo de índices permitido será igual ao menor número de palavras encontrado em 1 só dos 10 arquivos fornecidos (o menor número entre todos). Número esse que deve ser encontrado pelo programa ao ler os 10 arquivos. Um arquivo HashDocs.txt deve ser gerado com todos os valores (e.g. “abacaxi” chave K índice X), e a estatística gerada por sua função de espalhamento (“*hashing*”), ou seja, número total de palavras, número máximo do índice, total de colisões, número de elementos da maior lista de colisões. Além da saída em arquivo o programa deverá mostrar esses resultados em tela e interagir com o usuário pedindo uma palavra a ser encontrada e informando o seguinte: “Palavra não encontrada”, ou “Palavra presente nos arquivos x1, x2, etc”.

O código deve ser bem documentado, de forma modular com funções para cada tarefa independente, realizado por dois (2) estudantes do curso usando *“pair programming”*, e entregue via sistema http://aprender.unb.br do curso, no prazo estipulado.