public class ListaInvertida {

private PalavraNo[] tabela;

private int tamanho=0;

private int colisoes = 0;

public ListaInvertida(int tamMax) {

tabela = new PalavraNo[tamMax];

}

public int tamanho() {

return tamanho;

}

public PalavraNo[] getTabela() {

return tabela;

}

private int funcaoHash(String chave) {

int soma = 0;

for (int i = 0; i < chave.length(); i++) {

soma += chave.charAt(i);

}

return soma % tabela.length;

}

public PalavraNo busca(String chave) {

if (chave == null) {

return null;

}

for (PalavraNo retorno = tabela[funcaoHash(chave)];

retorno != null; retorno = retorno.getProx()) {

if (retorno.getPalavra().equals(chave)) {

return retorno;

}

}

return null;

}

public boolean insere(String palavra, String documento) {

if (busca(palavra) != null) {

int indice = funcaoHash(palavra);

if(tabela[indice].getPalavra().equals(palavra) &&

tabela[indice].buscaDocumento(documento)!=null){

return false;

}else if(tabela[indice].getPalavra().equals(palavra) &&

tabela[indice].buscaDocumento(documento)==null){

tabela[indice].insereDocumento(documento);

return true;

}

}

int indice = funcaoHash(palavra);

PalavraNo topo = tabela[indice];

if (tabela[indice] != null) {

colisoes++;

}

tabela[indice] = new PalavraNo(palavra, documento);

tabela[indice].setProx(topo);

tamanho++;

return true;

}

public double cargaUtilizacao() {

if(tamanho == 0) return 0.0;

double res;

int cont = 0;

for (int i = 0; i < tabela.length; i++) {

if(tabela[i]!=null) cont++;

}

res = (double)cont /(double)tabela.length;

res\*=(double)100;

return res;

}

public int getNumColisoes(){

return colisoes;

}

public double tamanhoMedioListas() {

if(tamanho == 0) return 0.0;

double cont = 0;

for(int i=0;i<tabela.length;i++){

if(tabela[i]!=null) cont++;

}

double res = (double)tamanho/(double)cont;

return res;

}

public void rehash(int novoTamanho){

tamanho=0;

colisoes=0;

String res ="";

PalavraNo []aux = tabela;

//for(int i=0;i<tabela.length;i++){

// aux[i]=tabela[i];

//}

tabela=null;

tabela=new PalavraNo[novoTamanho];

for(int i=0;i<aux.length;i++){

for(PalavraNo atual = aux[i]; atual!=null;atual=atual.getProx()){

for(DocumentoNo aux2 = atual.getDocumentosTopo(); aux2!=null;aux2=aux2.getProx()){

res+=aux2+" ";

}

//int indice = funcaoHash(atual.getPalavra());

insere(atual.getPalavra(),res);

res="";

}

}

for(int i=0;i<aux.length;i++){

if(aux[i]!=null){

int indice = funcaoHash(aux[i].getPalavra());

tabela[indice]=aux[i];

}

}

}

public String toString(){

String res="";

int cont=0;

for(int i=0;i<tabela.length;i++){

for (PalavraNo atual = tabela[i]; atual != null;

atual = atual.getProx()){

res+="[Indice "+cont+"] ";

res+="Palavra '";

res+=atual.getPalavra();

res+="': ";

for(DocumentoNo aux = atual.getDocumentosTopo(); aux!=null;aux=aux.getProx()){

res+=aux+" ";

}

res+="\n";

}

cont++;

}

return res;

}

public void imprime() {

for (int i=0; i<tabela.length; i++){

for (PalavraNo atual = tabela[i]; atual != null; atual = atual.getProx()){

for (DocumentoNo aux3 = atual.getDocumentosTopo(); aux3 != null; aux3 = aux3.getProx()){

System.out.println(atual.getPalavra()+ " = " + aux3);

}

}

}

}

}