



Trabalho Recuperação – Sumário

SCC-502 Algoritmos e Estruturas de Dados 1 M. Cristina/Jorge

1 Introdução

O trabalho deverá ser feito **individualmente** e submetido para o sistema Run.Codes.

Data de entrega: 09 de fevereiro de 2015.

1.1 Critério de Avaliação

$TRec$ = Nota do trabalho.

$N_{acertos}$ = Número de casos corretos no Run.Codes.

$Organizacao$ = Organização e modularização do código.

$Operacoes$ = Operações nos TAD Árvore e Lista.

$$TRec = N_{acertos} * 0.7 + Organizacao * 0.1 + Operacoes * 0.2$$

2 Descrição do Trabalho

Seu programa deve, a partir de um conjunto de palavras e da página onde elas aparecem, permitir a consulta e a impressão de um sumário.

O sumário deve apresentar a listagem das páginas em ordem crescente e, para cada página, as palavras que aparecem na página **em ordem alfabética**.

2.1 Requisitos

Implemente o índice como uma árvore de busca binária.

Implemente a lista de palavras como uma lista simplesmente encadeada ordenada.

Utilize as seguintes estruturas:

```
typedef struct tL{
    struct tL *prox;
    char palavra[50];
}ListaPalavra;

typedef struct tNo{
    struct tNo *esq, *dir;
    int numero_pagina;
    ListaPalavra *lista;
}NoArvore;
```

3 Entrada

A entrada deve ser lida do teclado (`stdin`).

A entrada contém apenas um caso de teste, composto por várias linhas. Cada linha tem início com um comando: **inserir**, **buscar**, **listar** e **sair**.

3.1 inserir

O comando inserir tem o seguinte formato:

i <palavra> <pagina>

em que <palavra> é uma string que contém apenas os caracteres [a ... z] (não é sensível a caixa | "a" == "A") e <pagina> é o número da página em que a palavra aparece.

<pagina> ∈ {1, 2, 3, 4, 5, ...1000}.

| <palavra> | ≤ 50

Atenção: Palavras repetidas em uma mesma página devem ser desconsideradas.

3.2 buscar

O comando buscar tem o seguinte formato:

b <pagina>

E deve imprimir as palavras que aparecem na página <pagina> separados por quebra de linha.

3.3 listar

O comando listar tem o seguinte formato:

l

e deve imprimir o sumário ordenado por ordem alfabética.

3.4 sair

O comando tem o formato:

s

e deve desalocar a memória da árvore, das listas e sair do programa.

4 Saída

A saída deve ser apresentada na saída padrão, através do comando `printf`

Apenas os comandos **buscar** e **listar** devem apresentar algo na tela.

4.1 buscar

Suponhamos que a página 2 contém as palavras "pera", "uva" e "maca".

Se o usuário digitar o comando

```
b 2
```

O programa deve imprimir:

```
maca
pera
uva
```

Caso a página não contenha nenhuma palavra cadastrada, o programa não deve imprimir nada.

4.2 listar

O comando **listar** deve imprimir o sumário em ordem alfabética. Como a estrutura utilizada é uma árvore binária de busca, isto equivale a realizar um percurso em-ordem na árvore.

O programa deve imprimir, separados por quebra de linha,

```
<pagina>
<palavra 1>
<palavra 2>
...
<palavra n>
<pagina>
<palavra 1>
...
<palavra n>
```

Atenção: Só existe uma quebra de linha entre as palavras.

5 Exemplo de execução

Entrada

i abobora 1
i banana 3
i fruta 5
i fruta 1
i abacate 1
i abacate 1
i fruta 2
i melancia 4
i banana 2
b 1
b 2
b 3
l
s

Saída

abacate
abobora
fruta
banana
fruta
banana
1
abacate
abobora
fruta
2
banana
fruta
3
banana
4
melancia
5
fruta

6 Plágio

ATENÇÃO: O sistema Run.Codes utiliza a ferramenta MOSS, de Stanford, para detecção de plágio. Para mais informações, visite: [<http://theory.stanford.edu/aiken/moss/>](http://theory.stanford.edu/aiken/moss/).