MAC323/BCC PRIMEIRO EXERCÍCIO-PROGRAMA

PALAVRAS MAIS FREQÜENTES

Y. KOHAYAKAWA

Data de entrega. Você deve entregar este EP até as 18:00 do dia 30/3/2001, na secretaria do Departamento de Ciência da Computação, sala 256, bloco A.

1. Introdução

Neste primeiro exercício-programa, você se familiarizará com a noção de *tabela de símbolos.* Nesta etapa, você não precisará usar implementações sofisticadas de tais tabelas; a idéia principal é você entender como podemos usar tais estruturas. A referência básica para este EP é a parte inicial do Capítulo 12 de Sedgewick (inclusive §12.1).

2. O problema a ser resolvido

O programa que você escreverá neste exercício deve resolver o seguinte problema: (i) a entrada de seu programa é formada por um inteiro n e um texto e (ii) a saída de seu programa deve ser a lista das n palavras mais freqüentes no texto, com suas respectivas freqüências (número de ocorrências na entrada).

Observações. Uma *palavra* é uma seqüência maximal de letras. Você deve distinguir letras maiúsculas de minúsculas, de forma que `Vitória` e `vitória` devem ser consideradas palavras diferentes. As palavras que ocorrem um mesmo número de vezes devem ser listadas em ordem 'alfabética'. Use a ordem alfabética induzida pela tabela ASCII, por exemplo, `Zoroastro` deve vir antes de `alabastro`. O seu programa deve parar após n palavras terem sido devolvidas, caso a entrada tenha pelo menos n palavras.

Execução. A entrada de seu programa deve vir do `stdin` e o seu programa deve receber o inteiro n na linha de comando. A saída deve ser enviada para o `stdout`. Por exemplo, executando o seu programa no texto do *King James' Bible* do Projeto Gutenberg (removendo o cabeçalho contendo informações sobre o Projeto Gutenberg, termos de distribuição, etc), obtemos algo como

```
meu_prompt > ep1 -n5 < entrada.txt
     62064 the
     38847 and
     34428 of
     13378 to
     12846 And
```

Esta saída diz que, por exemplo, a palavra mais freqüente é *the*, com 62064 ocorrências.

Date: Versão de 2 de abril de 2001.

3. Estrutura geral do programa

Você deve estruturar o seu programa da seguinte forma.

3.1. **Objetos das tabelas de símbolos.** Os objetos a serem armazenados em sua tabela de símbolos devem ser do tipo `Item`, implementados em `Item.c` e manipulados através da interface `Item.h`.

3.2. **Tabela de símbolos.** A sua tabela de símbolos deve ser implementada em `ST.c` e o acesso a ela deve ser *estritamente através da interface* `ST.h`. Projete a interface `ST.h` de forma que ao alterar a implementação da tabela de símbolos, você não precisará alterar outras partes do programa. Possivelmente, você disponibilizará uma função de protótipo

```
void ST_first_n(void (*visit)(Item), int n);
```

em `ST.h`, para enviar as n palavras mais freqüentes à saída, por exemplo, com a chamada

```
ST_first_n(ITEMshow, n);
```

em seu programa principal.

Procure implementar sua tabela de símbolos de forma que o seu programa possa manipular arquivos de tamanho 'razoavelmente grandes'. Com o que vimos em sala até o momento, provavelmente não sabemos como escrever um programa que possa realmente processar um arquivo como o *King James' Bible*, que contém algo como 800.000 palavras.

3.3. **Arquivos de teste.** Uma boa fonte de arquivos para testar o seu programa é o Projeto Gutenberg.

4. Observações

1. *Este EP é individual.*
2. Preste atenção na modularização descrita neste enunciado. Se você seguir estas instruções, será mais fácil você fazer o EP2.
3. Seja cuidadoso com sua programação (correção, documentação, apresentação, clareza do código, etc), dando especial atenção a suas estruturas de dados. A correção será feita levando isso em conta.
4. *Não imponha restrições arbitrárias sobre a entrada.* Quando usado forma inesperada, o seu programa deve parar de forma 'graciosa'.
5. Organizem-se na turma para fazer bons testes. Comparem entre vocês o desempenho de seus programas.
6. Seja criativo. Se seu programa fizer algo a mais do que foi pedido você poderá ganhar algum bônus na nota. Por outro lado, você *deve* respeitar as especificações dadas neste enunciado.
7. Entregue o seu EP seguindo os moldes usuais: cabeçalho claro, disquete com os arquivos (coloque um `Makefile`) e eventuais arquivos de teste.

Observação final. Enviem dúvidas para a lista de discussão da disciplina. Eventualmente, ajustes no enunciado ocorrerão ao discutirmos este EP na lista.