MAC–IME–USP

Carlos Eduardo Ferreira
Sala 108C Tel.: 3091 6079
e-mail cef@ime.usp.br
monitor Guilherme Puglia
e-mail puglia@ime.usp.br

MAC 323 – Estruturas de Dados

Primeiro semestre de 2011

Árvores balanceadas – **Entrega: 29 de maio de 2011**

Como todos sabem, Fabuloso é um excelente jogador de futebol. Além disso, nas horas vagas, ele gosta de estudar computação. E, como é comum no futebol, tem vários amigos em todos os times.

Nos últimos dias surgiu uma aposta entre os jogadores para saber quem era o mais popular na internet. Entretanto, como ninguém tem ideia de como solucionar este problema foram consultar o Fabuloso.

O Fabuloso então sugeriu que fosse coletado um conjunto de textos e que estes fossem processados por um programa de computador. É aí que você entra.

Basicamente, sua tarefa neste exercício será: a partir de um texto dado, construir uma estrutura de dados eficiente para fazer buscas por palavras que têm tamanho em um intervalo dado. Estas palavras deverão ser mostradas em ordem de frequência com que ocorrem, e para cada palavra deve ser fornecida a frequência com que ocorre.

A ideia é "filtrar" os resultados de modo que seja possível limitar o tamanho mínimo e o tamanho máximo das palavras que serão exibidas a cada consulta.

Note que seu programa deve ser o mais eficiente possível. Para ajudá-lo, o Fabuloso tem algumas dicas de como fazer isso.

Inicialmente, dada uma palavra p, precisamos saber se p já ocorreu anteriormente no texto. Uma forma eficiente, e que Fabuloso deseja que você implemente, é construir uma árvore de busca binária balanceada (AVL ou rubro-negra, você escolhe) para armazenar as palavras distintas que ocorreram até o momento.

Depois de saber se a palavra já ocorreu ou não no texto, precisamos atualizar sua frequência. Isso pode ser feito em tempo constante.

Agora, as consultas serão dadas por pares de inteiros, que vão ser o tamanho mínimo e máximo das palavras desejadas, e sua tarefa será listar as palavras que ocorrem com tamanho neste intervalo em ordem alfabética, mostrando para cada uma a frequência com que ocorrem.

Para realizar esta última tarefa de forma eficiente, Fabuloso sugeriu uma estrutura abaixo. A ideia básica é criar uma árvore binária onde cada nó representa um intervalo $[a, b]$. Assim, se olharmos para uma folha temos que $a = b$ (ou seja, o intervalo tem apenas um elemento). Em cada nó desta árvore teremos um ponteiro para uma estrutura que conterá todas as palavras cujo tamanho está no intervalo representado pelo nó. Perceba que uma palavra pode pertencer a várias destas estruturas.

Especificação da Entrada

A entrada consiste de somente um texto e um conjunto de consultas. A primeira linha contém um inteiro **N** que é o número de linhas do texto. As próximas N linhas contêm o texto de entrada. O texto só contém caracteres a-z, A-Z, ',', '.', ':' e espaços.

Em seguida, segue em uma linha um inteiro **M** que é o número de consultas que desejamos fazer. Cada uma das próximas M linhas contém dois inteiros que são, respectivamente, a quantidade mínima e a quantidade máxima de letras desejadas.

Especificação da Saída

Para cada consulta, deve ser impressa uma linha contendo a string "Contém K elementos:" onde K é a quantidade de palavras que serão impressas. As próximas K linhas serão da seguinte forma: "[F] palavra", onde F é a frequência da palavra, e as K palavras deverão estar ordenadas em ordem crescente pela frequência, e, em caso de empate, pela ordem alfabética. Após cada caso de teste, imprima uma linha em branco.

Exemplo de Entrada

```
4
teste aaa a bbb teste tres
aaa
cinco
seisss a a
5
0 10
1 1
2 2
3 3
3 5
```

Exemplo de Saída

```
Contém 7 elementos:
[1] bbb
[1] cinco
[1] seisss
[1] tres
[2] aaa
[2] teste
[3] a

Contém 1 elementos:
[3] a

Contém 0 elementos:

Contém 2 elementos:
[1] bbb
[2] aaa

Contém 5 elementos:
[1] bbb
[1] cinco
[1] tres
[2] aaa
[2] teste
```