

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
ALGORITMOS E ESTRUTURAS DE DADOS III

1º semestre de 2016

Professor: Leonardo Chaves Dutra da Rocha

Trabalho Prático 3

Data de Entrega: 20 de Maio 2015.

Trabalho Dupla

Este trabalho prático tem por objetivo exercitar conceitos e práticas dos algoritmos de processamento de caracteres.

Autocompletar

Como um bom aluno de Ciência da Computação, tenho certeza que já ficou muito curioso em saber como funciona o *autocomplete* da ferramenta de busca do Google. Por exemplo, você começa a digitar *New* e surgem para você diversas opções. como mostrado na figura abaixo:

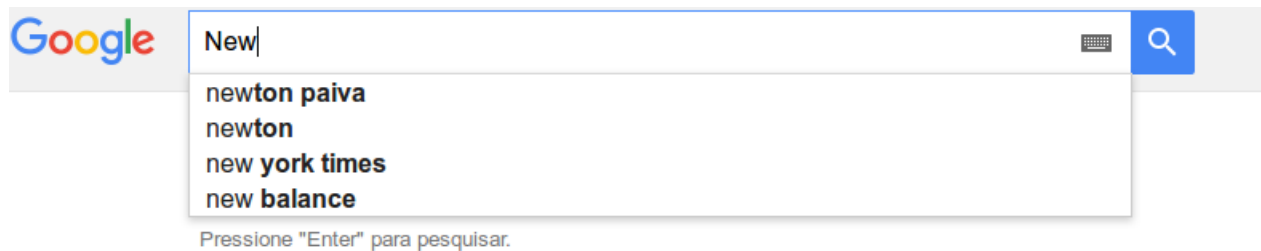


Figura 1: Exemplo de Autocompletar

Outros exemplos de aplicações que usam isso são: Internet Movie Database (IMDB), que usa função de autocompletar para exibir os nomes de filmes a medida que o usuário digita; telefones celulares também tem essa função implementada para acelerar a entrada de texto.

Como sou um professor muito bacana, resolvi então dar a oportunidade para vocês investigarem como isso funciona. Assim, nesse trabalho você deve escrever programas que implementem essa função de autocompletar. Você receberá como entrada um conjunto de N palavras (dicionário), sendo que cada palavra virá acompanhada de um peso, que corresponde ao número de vezes que a mesma já foi utilizada no passado. Além disso, você receberá o prefixo de uma palavra, ou parte de uma expressão como mostrado na figura acima, bem como um inteiro k . Assim, seu programa deverá retornar as k palavras que casam com o prefixo passado, ordenado de forma decrescente de acordo com o peso das palavras.

Basicamente você deverá gerar duas implementações. Uma utilizando um método força bruta e outra utilizando estratégias e estruturas de dados eficientes. O força bruta é simples de ser implementado, porém muito lento e não adequado para um ambiente real. Assim, você poderá utilizá-lo para comparar com sua proposta de solução. Além disso, você deverá implementar também uma interface gráfica para seu algoritmo. Essa interface pode ser Web ou para estação de trabalho, usando por exemplo java e swing.

Entrada e Saída

Um exemplo de arquivo de dicionários:

```
10
1000 Shanghai
900 Buenos Aires
750 Mumbai, India
700 Mexico City
650 Karachi
600 İstanbul
500 Delhi
450 Manila
400 Moscow
340 Manaus
```

Exemplo de um prefixo e considerando $k = 3$ e sua respectiva saída:

```
M
Mumbai
Mexico
Manila
```

Observações:

- O código fonte do trabalho deve ser submetido para compilação e execução em ambiente Linux, tendo como padrão os computadores dos laboratórios do DCOMP.
- Deve ser escrito na linguagem C (trabalhos implementados em outras linguagens como C++/Java/Python e outras não serão aceitos);
- As estruturas de dados devem ser alocadas dinamicamente e o código deve ser modularizado utilizando os arquivos .c .h.
- O utilitário Make deve ser utilizado para compilar o programa;
- A saída deve ser impressa no arquivo pedido seguindo estritamente o formato da especificação caso contrário o resultado será considerado errado;
- Faça seu código de forma legível

Avaliação

Deverão ser entregues:

- listagem das rotinas;
- documentação contendo;
 - descrição das soluções e estruturas de dados utilizadas;
 - análise da complexidade das rotinas;
 - análise dos resultados obtidos.
 - a documentação não pode exceder 12 páginas.

Distribuição dos pontos:

- execução (E)
 execução correta: 80%
- estilo de programação
 código bem estruturado: 10%
 código legível: 10%
- documentação (D)
 comentários explicativos: 30%
 análise de complexidade: 10%
 análise de resultados: 60%

A nota final é calculada como a média harmônica entre execução (E) e documentação (D):

$$\frac{D * E}{\frac{D+E}{2}}$$