

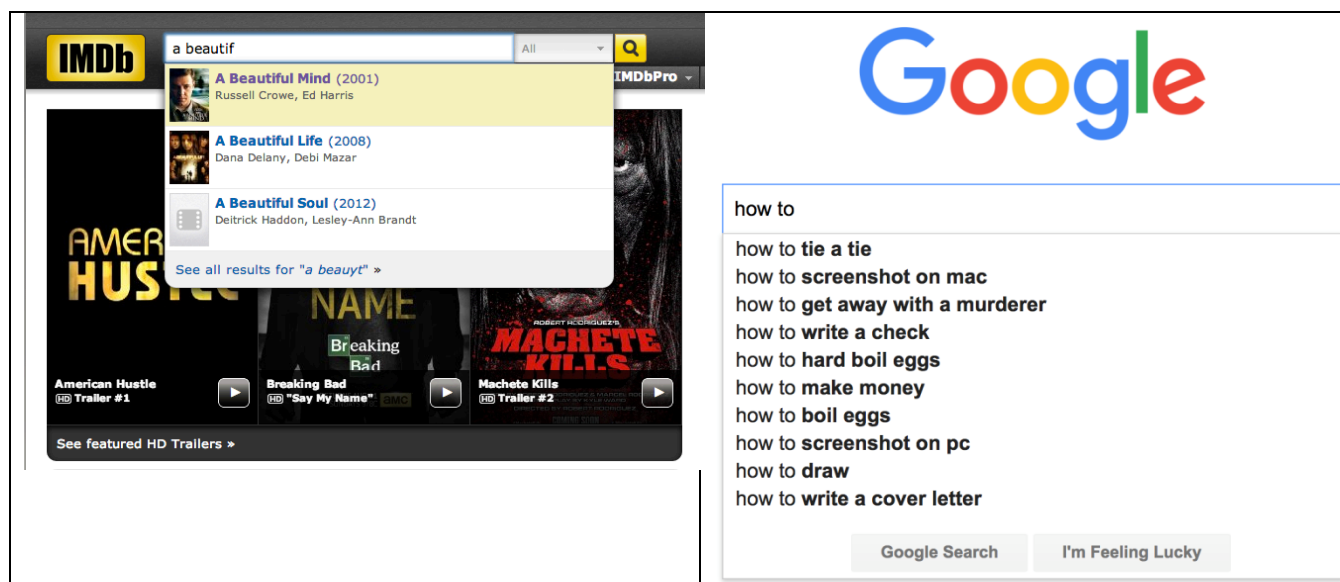
TRABALHO FINAL Autocompletar

1 Objetivo

Este trabalho tem por objetivo proporcionar aos alunos a oportunidade de aplicar os conhecimentos adquiridos e as estruturas de dados desenvolvidas em aula na solução de um problema que utilize várias dessas estruturas. O trabalho prático envolve a utilização de estruturas de dados do tipo listas e árvores.

2 Especificação da Aplicação

A funcionalidade de autocompletar é muito usada em motores de busca na Web, serviços de mensagem de celular e editores de código. Ele consiste em tentar prever quais serão os próximos caracteres ou palavras que o usuário irá digitar, à medida que o ele digita. Veja os exemplos abaixo:



Nesses exemplos, o aplicativo prevê a probabilidade do usuário digitar cada consulta e apresenta ao usuário uma lista de sugestões em ordem decrescente de peso. Esses pesos são determinados por dados históricos, como a arrecadação da bilheteria para filmes, frequências de consultas submetidas ao motor de busca ou o histórico de digitação de um usuário de telefone celular.

O desempenho da funcionalidade de autocompletar é crítico em muitos sistemas. Por exemplo, considere um mecanismo de pesquisa que executa um aplicativo de preenchimento automático. De acordo com estudos, o aplicativo tem apenas cerca de 50ms para retornar uma lista de sugestões para que ele seja útil para o usuário. Além disso, em princípio, ele deve executar este cálculo para cada tecla digitada na barra de pesquisa e para cada usuário!

A tarefa a ser desenvolvida simula, de forma bastante simplificada, o funcionamento de um mecanismo de sugestão de palavras. Você deve projetar estruturas de dados e algoritmos para resolver o problema de forma **eficiente**.

3 Definições:

Uma *consulta* é uma sequência de caracteres que representa o prefixo de zero ou mais palavras. Por exemplo, se o usuário digita o prefixo/consulta “*car*” o sistema deve sugerir palavras que começam com este prefixo tais como “*carro*”, “*carta*”, *etc.* em ordem decrescente de *peso*. O peso é um número inteiro atribuído a cada palavra que reflete a sua frequência (*i.e.* quanto mais frequente, maior o peso). Os pesos das palavras serão fornecidos. Diferenças entre letras maiúsculas e minúsculas devem ser desprezadas (ex: a = A).

As entradas e saídas da aplicação são:

- Entradas: (i) arquivo texto as palavras e seus pesos; (ii) arquivo texto com as consultas/prefixos (uma consulta por linha) e (iii) número de sugestões desejadas
- Saídas: (i) arquivo com a consulta e as sugestões de palavras completas *em ordem decrescente de peso*; e (ii) tempo gasto no processamento.

4. Exemplo de funcionamento

Comando: C:\minhaaplicacao wikitionary.txt consulta.txt saida.txt 5

Entrada (i) arquivo com palavras e seus pesos: wikitionary.txt

```
5627187200 the
3395006400 of
2994418400 and
2595609600 to
1742063600 in
1176479700 i
1107331800 that
1007824500 was
879975500 his
...
392323 calves
```

Entrada (ii) arquivo com as consultas: consultas.txt

```
auto
comp
the
xyz
```

Saídas arquivo com as consultas, sugestões e tempo gasto: saida.txt

```
auto
    619695 automobile
    424997 automatic
comp
    13315900 company
    7803980 complete
    6038490 companion
    5205030 completely
    4481770 comply
the
    5627187200 the
    334039800 they
    282026500 their
    250991700 them
    196120000 there
xyz
    nenhum resultado encontrado

Tempo: 0.01ms
```

5 Requisitos

- Escolher/propor estruturas de dados adequadas para o processamento *eficiente* da tarefa utilizando listas e ou árvores.
- Redigir um relatório que explique com *riqueza de detalhes* as estruturas escolhidas/propostas defendendo as escolhas feitas.
- A aplicação deve ser chamada **a partir da linha de comando** (passando parâmetros para o main).
- O trabalho deve ser feito, preferencialmente, em duplas. A linguagem de programação aceita é C (Não é C++ nem C#).

6. Entrega e Apresentação

- **20 de julho de 2017 (10% bônus)** apresentação (no horário da aula) e entrega pelo Moodle
- **25 de julho de 2017** apresentação (no horário da aula) e entrega pelo Moodle

7. Critérios de Avaliação

O trabalho deve ser realizado em duplas e deverá ser apresentado e defendido na data prevista.

Para a avaliação serão adotados diversos critérios:

- funcionamento,
- organização e documentação do código.
- tempo gasto no processamento das consultas;
- justificativa para escolha das estruturas de dados envolvidas; e
- relatório;

Importante:

Este trabalho deverá representar a solução da dupla para o problema proposto. O plágio é terminantemente proibido e a sua detecção incorrerá na divisão da nota obtida pelo número de alunos envolvidos. Para detectar o plágio, usaremos o software MOSS (<http://theory.stanford.edu/~aiken/moss/>).