Trabalho Alg III 2016-1

Andrey Ricardo Pimentel e Marcos Didonet Del Fabro

Primeiro Semestre de 2016

1 Introdução

A banda Vingadora está investindo em sua carreira, após lançar musicas de grande sucesso no Carnaval de 2016. Para lançar novas músicas, eles precisam complementar seu repertorio musical com mais músicas. O letrista da banda é muito esquecido para lembrar das palavras que gostaria de usar nas músicas.

Ele só que compor em inglês e sempre lembra de palavras parecidas mas nunca da palavra certa. O Violinista da banda, Daniel, viu que poderia ajudar seu letrista a encontrar palavras parecidas com a que estava pensando utilizando estruturas de dados conhecidas como arvores digitais. A arvore digital que Daniel lhe pediu para usar para catalogar palavras foi a TRIE.

O problema a ser resolvido é de computar a distância de edição simplificada (edit distance) entre duas strings: o número mínimo de inserções de caracteres simples, deleções ou substituições necessárias para converter uma string para outra.

Por exemplo, a distância de edição entre "Hello" e "Jello" é 1. A distância de edição entre "good" e "goodbye" é 3. A distância de edição entre qualquer palavra e ela mesma é 0.

A distância de edição pode ser usada para propósitos tais como sugerindo, em um corretor ortográfico, uma lista de substitutos plausíveis para uma palavra incorreta. Para cada palavra não encontrada no dicionário (e, portanto, presumivelmente com erros ortográficos), liste todas as palavras no dicionário, que estão à uma pequena distância de edição a partir do erro de ortografia (maximo 3).

2 Especificações

Daniel é exigente e esta anotando instruções especificas do que quer fazer Vamos ver abaixo o que ele quer fazer:

O programa deverá ler um arquivo com palavras chamado ***.txt e criar uma árvore TRIE com essas palavras. As palavras 1) não estão ordenadas alfabeticamente, 2) as palavras com acento e caracteres especiais (não letras) poderão ser ignoradas, 3) não há distinção entre maiúsculas e minusculas, 4) o

tamanho do arquivo não é fixo. Um exemplo de arquivo de entrada pode ser encontrado em: `http://www.ime.usp.br/~ueda/br.ispell/pt_BR.aff.gz`

O programa deverá ler um arquivo com as consultas, chamado consultas.txt. Cada consulta é composta de uma palavra e um número que é a distância de edição. O número máximo a ser usado será 3.

Para cada consulta, deverá realizar uma busca na TRIE, encontrar as palavras que estão a uma determinada distância de edição da palavra original e listá-las separadas por vírgula e na mesma linha. Listar no máximo 20 palavras.

O programa terá o nome de "dicionario" e será feito em linguagem C.

O trabalho poderá ser feito em equipes de até 2 (duas) pessoas.

3 Exemplo de execução

Exemplo de execução

- Arquivo de entrada com palavras do dicionario:

  adriatico
  adroaldo
  afeganistao
  Alasca
  Alberta
  Albertina
  boiaras
  boiardes
  boiarei
  boiareis
  boiarem
  boiaremo

- Arquivo de entrada com palavras a serem testadas:

  adriatico 2
  alberta 3
  aberta 2
  boiada 2
  boiada 3

- Arquivo de saída com o resultado:

  adriatico:Adriatico
  alberta:Alberta, Albertina
  aberta:Alberta

boiada:boiaras
boiada:boiaras,boiarei,boiarem

4 Entrega

O trabalho deve ser entregue até 23h59m do dia 08 de junho de 2016, contendo como título do e-mail "CI057 - Trabalho.2016-1" para o professor de sua turma:

- andrey@inf.ufpr.br
- marcos.ddf@inf.ufpr.br

A data da entrega do trabalho será até as 23h59m do dia 08/06/2016. Entregas fora do prazo terão desconto na nota por dia de atraso.

5 Detalhes de Entrega

- Deve ser enviado um arquivo compactado tar.gz com, no minimo, os seguintes arquivos:
  - main.c
  - dicionario.c
  - dicionario.h
  - Makefile
  - LEIAME (detalhes do trabalho que achar interessante, dificuldades que teve na implementação e bugs conhecidos.)
- o trabalho deverá poder ser compilado no ambiente computacional do dinf.
- a compilação deverá ser feita com make.
- O arquivo Makefile deve possuir opção clean (apaga todos os arquivos objeto .o)
- o trabalho deverá ser executado com a seguinte linha de comando

```
.\dicionario < entrada.txt > saida.txt
```

- a verificação do resultado será feita com

```
diff saida.txt saida_padrao.txt
```

- O compactado deve possuir o nome como login.tar.gz. Em caso de trabalho em grupo, deve possuir o login de todos - login1-login2.tar.gz . Apos descompactar deve gerar uma pasta com o nome do compactado (login ou login1-login2) com todos os arquivos necessários, sem subdiretórios.

Qualquer duvida, não hesitem em procurar a monitoria no horário disponível ou mandar e-mail (abovs14@inf.ufpr.br).