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

Distância entre palavras – Entrega: **26 de junho**

Várias experiências podem ser feitas com dicionários de palavras. Neste exercício-programa faremos uma dessas experiência: vamos encontrar caminhos "mais curtos" que ligam as palavras. Dizemos que duas palavras do dicionário são vizinhas se uma puder ser transformada na outra através de uma das operações abaixo:

- remoção de uma letra: o custo da operação é 1 se a letra removida for uma vogal e 2 se for uma consoante. Por exemplo, `alta` e `ata` são vizinhas.
- troca de letras de uma mesma palavra: o custo da operação é 1 se ambas as letras são vogais, 2 se são duas consoantes e 3 se uma for vogal e a outra consoante. Por exemplo, `nato` e `nota` são vizinhas, assim como `nota` e `tona`.
- inserção de uma letra: o custo da operação é 1 se a letra inserida for uma vogal, e 2 se for uma consoante. Por exemplo, `toda` e `toada` são vizinhas, assim como `alta` e `altar`.
- substituição de uma letra: o custo é 1 se ambas as letras são vogais, 2 se são ambas consoantes e 3 se uma é vogal e a outra é consoante. Por exemplo, `gato` e `rato` são vizinhas, assim como `prata` e `preta`.

Usando estas operações podemos construir caminhos entre palavras de um dicionário. Por exemplo, se o dicionário é formado pelas palavras:

```
gato, pato, lata, nata, nato, tona, nota, alta, ata, preta, ato, pata, preto,
rato, prata, prato, catarata, carta.
```

As palavras `alta` e `preto` estão ligadas pelo seguinte caminho de palavras no dicionário:

```
alta -> lata -> nata -> nato -> pato -> prato -> preto -> preta
```

O custo do caminho é a soma dos custos das operações realizadas. No caso, 3+2+1+2+2+1+1 = 12. Neste exercício programa será dado um dicionário de palavras e vários pares de palavras e você deverá encontrar o caminho de palavras com menor custo ligando as palavras dadas (se existir). Para resolver este problema você deverá implementar as seguintes estruturas:

- uma tabela de hash para armazenar as tabelas do dicionário e fazer as buscas eficientemente;

- uma fila de prioridade (heap) para implementar o algoritmo que encontra o caminho de menor custo entre duas palavras.

Você pode assumir que todas as palavras são formadas por letras minúsculas, sem acentos, a maior palavra possível tem 20 letras e o dicionário pode ter até 10000 palavras.

A entrada do programa é dada por inteiro n seguido por n palavras do dicionário, uma por linha. Em seguida é dado um inteiro m e nas m linhas seguintes são dados pares de palavras, um por linha, para os quais se deseja encontrar um caminho mínimo como descrito anteriormente. Na saída você deverá mostrar o caminho, quando existir, e o seu custo. Veja o exemplo abaixo.

Você deverá ter uma estrutura para armazenar as vizinhanças de cada uma das palavras do dicionário. A construção dessa estrutura poderá ser demorada. Pense em como fazê-la da forma mais eficiente possível.

Exemplo de entrada

```
18
gato
pato
lata
nata
nato
tona
nota
alta
ata
preta
ato
pata
preto
rato
prata
prato
catarata
carta
2
alta preta
carta catarata
```

Exemplo de saída

```
O menor caminho entre alta e preta tem custo 8
alta -> lata -> pata -> prata -> preta

Não existe caminho entre carta e catarata
```