

{TP2}

Sugestões Doodle

Algoritmos e Estruturas de Dados III

22 de Maio de 2018

1 Introdução

O Doodle é um serviço de busca de conteúdo pela internet com grande base de usuários. Você foi contratado pela Doodle para desenvolver um algoritmo que irá auxiliar seus usuários com problemas de digitação, o que é algo bastante comum entre os pessoas que digitam rapidamente.

A ideia é implementar um sistema de recomendação de palavras parecidas com o que foi buscado, dado um limite de mudanças. Para isso, você recebe um conjunto de palavras D que serve como dicionário. Essas palavras fazem a indexação do conteúdo do Doodle. Além disso, você recebe uma palavra de consulta q e recebe um inteiro n , informando o máximo de operações para transformar uma palavra fonte em uma palavra destino. As seguintes operações são válidas e tem custo unitário:

1. **Inserção de caractere:** um caractere qualquer pode ser inserido na palavra de consulta.
2. **Remoção de caractere:** um caractere qualquer pode ser removido da palavra de consulta.
3. **Substituição:** um caractere qualquer da palavra de consulta pode ser alterado por outro caractere.

Sua função é retornar todas as palavras do dicionário que podem ser reconstruídas a partir da palavra de consulta, aplicando no máximo n operações. A sua saída deve mostrar as palavras que exigem menos alterações antes de palavras que exigem mais alterações. Caso 2 ou mais palavras exijam a mesma quantidade de alterações, você precisa desempatar usando a ordem léxica (i.e. alfabética).

2 Instruções e dicas

1. A implementação desse problema pode requerer utilizar um dos paradigmas de programação visto em sala (guloso, programação dinâmica ou divisão e conquista). A utilização de pelo menos um dos paradigmas é obrigatória.
2. As palavras de consulta nem sempre estarão incorretas.
3. No moodle estão 10 arquivos de teste de tamanho crescente para você verificar o seu código. Para ser corrigido, se código deve passar em testes **do tamanho** do teste 6.
4. A solução de força bruta para esse problema pode ser bastante lenta quando as entradas passam a ser médias ou grandes.
5. Algumas palavras do dicionário ou da consulta podem não fazer sentido nenhum em idiomas conhecidos.
6. As entradas não possuem espaço. Dessa forma, tanto a palavra de consulta, quando cada palavra no dicionário é uma sequência de caracteres alfanuméricos.

3 Exemplo com Entrada e Saída

O formato de entrada é o seguinte:

```
1      D      // numero de entradas no dicionario
2      n      // numero de mudancas aceitas
3      q      // Palavra de consulta
4      d0     // Primeira palavra do dicionario
5      d1     // Segunda palavra do dicionario
6      ...
7      d(D-1) // (d-1)-esima palavra do dicionario
```

Sua função é mostrar todas as palavras que podem ser obtidas com no máximo n operações, em ordem crescente no número de alterações. Por exemplo, para:

```
1      9
2      3
3      borracha
4      boracha
5      pia
6      Bicicleta
7      bucha
8      barraca
9      borracha
10     braca
11     lima
12     trape
```

A saída esperada será:

```
1      borracha
2      boracha
3      barraca
4      braca
```

4 O que entregar

Você deve submeter um arquivo compactado `seu_nome_sua_matricula.tar.gz` contendo¹:

1. Todos os arquivos do código `.c` e `.h` que foram implementados.
2. Um arquivo `makefile` **que crie um executável `tp2`**,
3. Sua documentação.

Sua documentação deve ter até 10 páginas contendo:

1. Uma breve descrição do problema.
2. Explicações das estruturas de dados e dos algoritmos utilizados para resolver o problema.
3. Análise da complexidade da solução proposta (espaço e tempo).
4. Um experimento com o tempo gasto por execução para os diferentes arquivos teste (podem ser os exemplos para teste).

¹Para gerar isso no linux usa-se o comando `tar`. No windows, use o 7-Zip (<https://www.7-zip.org/>)

Seu trabalho prático receberá o arquivo de entrada na entrada padrão, e será executado da seguinte forma (supondo que a entrada esteja num arquivo `entrada1.txt`, e que meu arquivo chame `caio_nepo_123456.tar.gz`):

```
1 tar -xvzf caio_nepo_123456.tar.gz
2 make -C ./caio_nepo_123456
3 ./caio_nepo_123456/tp2 < entrada1.txt
```

O seu TP deverá ser entregue de acordo com a data especificada no moodle. A penalidade em porcentagem para os TPs atrasados é dada pela fórmula $2^{(d-1)}/0.16$.