Trabalho - Dicionário de Anagramas

Prof. Dr. Juliano Henrique Foleis

Anagramas

Um anagrama de uma palavra p é definido como uma outra q palavra do dicionário D que satisfaz a seguinte restrição:

$$(q = Perm(p)) \wedge (p \in D) \wedge (q \in D)$$

tal que $Perm(p)$ é alguma permutação da palavra p e D é o dicionário.

Exemplos

Os anagramas da palavra *cebola* são *cebola* e *cabelo*.

Os anagramas da palavra *amor* são *amor*, *armo*, *oram*, *ramo*, *roam*, e *roma*. (sim, *roam* é uma palavra da lingua portuguesa.)

Descrição do Trabalho

Você deve implementar um dicionário de anagramas para palavras da lingua portuguesa em C. Seu programa deve receber uma única palavra pela linha de comando (sem acentos) e imprimir na saída a lista de todos os seus anagramas em ordem alfabética, incluindo a própria palavra de entrada. Cada anagrama deve ser impresso em sua própria linha. Desconsidere a diferença entre maiúsculas e minúsculas. Por exemplo, caso o usuário entre com *CEBOLA*, a saída deve ser *cebola* e *cabelo*.

A busca pelos anagramas deve ser realizada em $\Theta(\lg n)$, tal que n é o número de palavras no dicionário da lingua portuguesa. O pré-processamento deve ocorrer no máximo em tempo $\Theta(n \lg(n))$.

O dicionário de palavras é fornecido. Para evitar ter que lidar com UTF-8 em C, o dicionário não contém acentos.

Estruturas de Dados

Quando ordenadas por letras, palavras anagramas são representadas pela mesma string. Por exemplo, *ramo* e *roma* são anagramas, e quando ordenadas, ambas se tornam *amor*. Desta forma, podemos usar uma estrutura chave-valor para armazenar os anagramas de uma palavra. A chave é a palavra ordenada por letras, e o valor é uma lista das palavras que são anagramas daquela palavra. Assim, durante a inserção deve ser feita a pesquisa na estrutura pela palavra com as letras ordenadas (chave). Se encontrar, a palavra original deve ser inserida na lista correspondente, que é apontada pelo nó da estrutura. Caso a palavra ordenada não esteja na estrutura, um novo nó deve ser adicionado com uma lista contendo inicialmente somente a palavra original.

A principal estrutura de dados que deve ser usada é a árvore rubro-negra, que é uma estrutura de dados chave-valor capaz de encontrar qualquer chave em $\Theta(\lg n)$.

Dicas

- A busca ser realizada em $\Theta(\lg n)$ quer dizer que você não deve usar uma busca sequencial para pesquisar os anagramas!
- O dicionário fornecido (br.txt) tem todas as palavras da lingua portuguesa. Cada palavra está em uma linha terminada por `\n`. Ele está codificado em ASCII.

Avaliação

- **É obrigatório usar a árvore rubro-negra como a estrutura de dados usada para pesquisar os anagramas! Outras estratégias/estruturas não serão aceitas!**
- O trabalho pode ser feito individualmente ou em duplas.
- O trabalho deve ser implementado exclusivamente em linguagem C.
- **Não é permitido alterar o arquivo *br.txt*.**
- A organização do código em módulos e TADs será avaliada.
- Somente as seguintes bibliotecas podem ser usadas: *stdlib*, *stdio* e *string*.
- A equipe deve implementar todas estruturas na "mão".
- A entrega do trabalho deverá ser feita impreterivelmente até dia 16/08/2021. Trabalhos entregues após esta data serão desconsiderados.
- **Em caso de plágio o trabalho será anulado**. Lembre-se, copiar de outro grupo é plágio, tanto quanto copiar códigos prontos da internet. No caso de cópia de outro(s) grupo(s), a nota de todos os grupos será anulada! No caso de cópia de códigos prontos da internet sua nota será anulada. Em caso de plágio não há recuperação de nota.
- É permitido usar os códigos fornecidos pelo professor no *Github* e os códigos das aulas.
- O trabalho deve ser entregue em formato .tar.gz ou .zip contendo o código-fonte, um *Makefile*, um arquivo *alunos.txt* e um arquivo *instrucoes.txt*. O arquivo *alunos.txt* deverá conter o nome e RA do(s) aluno(s) da equipe. O arquivo *instrucoes.txt* deverá conter instruções adicionais sobre como compilar o programa, caso necessário.
- A nota do trabalho será *70%* a implementação e *30%* uma apresentação breve, que ocorrerá na semana da entrega. Na apresentação o professor pedirá que a equipe explique como o trabalho foi dividido entre os membros da equipe e como foi realizada a modularização. A equipe também fará uma breve demonstração do programa em funcionamento.
- Os trabalhos devem ser entregues via Moodle. Todos os alunos da equipe devem enviar o trabalho.

Bom Trabalho!