

Trabalho 1 — Anagramas

Programação Orientada a Objetos

2021

1 Descrição geral

1.1 Anagramas

Uma palavra é um *anagrama* de outra se ela é escrita com as mesmas letras em uma outra ordem. Por exemplo, os seguintes pares são anagramas:

pera e pare
carro, corar e corra
assopra, apossar, raposas e posaras

Se a palavra A é anagrama da palavra B , então B é anagrama de A . Também se A e B são anagramas e B e C são anagramas, então A e C são anagramas. Isso significa que palavras formam conjuntos de palavras que são todas anagramas umas das outras. No exemplo anterior, cada uma das linhas forma um conjunto de anagramas

1.2 Busca de conjuntos de anagramas

O seu trabalho será, dado um arquivo de entrada com uma lista de palavras, achar todos os conjunto de anagramas formados pelas palavras fornecidas, lembrando que um conjunto pode ter apenas uma palavra.

2 Descrição detalhada

2.1 Entrada

A entrada para o seu programa será uma lista de palavras em um arquivo, onde cada palavra estará em uma linha distinta. Esse arquivo será codificado em UTF-8. O nome do arquivo de entrada será dado por um argumento de linha de comando, e pode ser lido pelo seu programa na `string sys.argv[1]`.

2.2 Saída

A saída deve ser um arquivo de texto com os conjuntos de anagramas encontrados na entrada. Você não deve procurar anagramas das palavras fornecidas, mas apenas procurar quais das palavras fornecidas são anagramas umas das outras. Cada conjunto de anagramas deve ficar em uma linha diferente. As palavras de um conjunto devem ser separadas umas das outras por uma vírgula e espaço em branco. A ordem das palavras no conjunto deve ser de acordo com a ordenação

de `str` definida pelo Python. A ordem dos conjuntos nas linhas pode ser a que você julgar mais conveniente. Esse arquivo deve usar codificação UTF-8. O nome do arquivo de saída deve ser o nome do arquivo de entrada acrescentando a extensão `.ana`. Por exemplo, se o arquivo de entrada se chamar `words` então o de saída se chamará `words.ana`.

2.3 Maiúsculas e minúsculas

Maiúsculas e minúsculas devem ser consideradas iguais ao realizar as comparações. O arquivo de saída deve ter apenas letras minúsculas.

2.4 Exemplo

Dado o arquivo de entrada seguinte:

```
pera
carro
posaras
corra
raposas
pare
corar
apossar
assopra
cisco
```

O arquivo de saída deve ter as seguintes linhas:

```
apossar, assopra, posaras, raposas
carro, corar, corra
cisco
pare, pera
```

(sendo que as linhas podem estar em outra ordem).

3 Entrega

Entregar pelo Moodle o código Python em um arquivo `.py`, incluindo comentários explicando as suas decisões de implementação.