Lista 8 - Prog 01 2024

Emilio Vital Brazil

Entrega: 08 de julho 2024

Todas as questões abaixo serão consideradas usando a sintaxe **Python 3.10**.

1. Implemente uma função que receba duas listas de números ordenadas em ordem decrescente e retorne uma única lista contendo os números das duas listas de entrada, também em ordem decrescente. Seu algoritmo deve ter complexidade $O(n + m)$ no pior caso, onde n e m são os números de elementos das listas de entrada, respectivamente.

2. Há N itens de roupas dispostos em uma loja, com o i-ésimo item sendo do tipo D_i. Alguns itens podem ser do mesmo tipo que outros.

   Você está comprando roupas em uma loja, mas também gostaria de variar nas suas compras. Os N itens serão apresentados a você, um após o outro em ordem, e para cada um deles você o comprará desde que não seja do mesmo tipo que qualquer um dos K itens anteriores que você comprou. Escreva uma função com os seguintes requesitos:

   - Entrada:
     - Um inteiro N que representa o número total de itens.
     - Um inteiro K que representa a quantidade de itens anteriores que você deve considerar para não repetir.
     - Uma lista de N inteiros D, onde D_i representa o tipo do i-ésimo item.
   - Saída:
     - Um inteiro que representa o número total de itens que você comprará.

3. Escreva uma função que determine se uma string que possa conter os caracteres '(' e ')' é válida. Uma string é considerada válida se:

   Todo parêntese de abertura '(' tiver um parêntese de fechamento correspondente ')' e todo parêntese de fechamento ')' fechar um parêntese de abertura correspondente '('. A função deve retornar True se a string for válida, e False caso contrário.

4. Escreva um módulo chamado Polygons que permita armazenar polígonos representados como listas de pontos 2D e uma cor correspondente para

cada polígono. O módulo deve ter, além do construtor, mais dois métodos: um para salvar os polígonos em um arquivo não binário, outro para ler arquivos nesse formato.

Requisitos:

- Classe Point2D:
  - Atributos:
    - **coord**: Tuple[float, float]
  - Métodos:
    - *__init__(self, x: float, y: float):* Construtor que inicializa um ponto.
- Classe Polygon:
  - Atributos:
    - **points**: uma lista de Point2D.
    - **color**: uma string que representa a cor do polígono.
  - Métodos:
    - *__init__(self, points: List[Point2D], color: str):* Construtor que inicializa os pontos e a cor do polígono.
- Classe Polygons:
  - Atributos:
    - **polygons**: Armazena as instâncias da classe Polygon e seus nomes.
  - Métodos:
    - *add_polygon(self, polygon: Polygon, name: str)*: Adiciona um polígono à lista de polígonos.
    - *remove_polygon(self, name: str)*: Remove o polígono que tem o nome 'name' da lista de polígonos.
    - *save_to_file(self, filename: str)*: Salva os polígonos em um arquivo não binário. O formato do arquivo deve permitir a leitura e escrita dos pontos, cores e nomes dos polígonos.
    - *load_from_file(self, filename: str)*: Lê os polígonos de um arquivo salvo no formato especificado.

Quando o módulo for executado a partir da linha de comando, ele deve realizar um teste de todas as funções da classe e imprimir os resultados esperados.

5. Estenda módulo do item anterior para plotar os polígonos na tela nas cores correspondentes usando o módulo **turtle**. Ou seja, crie uma função *plot_polygons* que recebe um objeto da classe *Polygons* que plota as linhas que contornam os polígonos na tela nas cores correspondentes.

   Você também deve fornecer um script que utilize o módulo.

   Importante: O módulo **turtle** não será ensinado no curso. Você deve aprender a utilizá-lo consultando a documentação do Python.