

Módulo 7 - Laboratório prático – Exercícios

Em cada módulo, os exercícios assinalados com ♣ são mais importantes. Os exercícios extra apresentam treino extra ou aplicações importantes dos TAD do módulo. São exercícios cuja prática complementa os conceitos e ajuda a perceber como os aplicar.

Exercícios:

1. Defina uma classe Vertex, que guarda uma identificação de um vértice (um objeto, cujo valor por omissão deve ser None). Não esqueça de incluir um método ("getter") para devolver o atributo identificador do vértice.
2. Defina uma classe Edge, que guarda uma identificação de uma aresta (um tuplo com referências para os objetos que representam os seus vértices terminais) e um peso. Não esqueça de incluir métodos para devolver estes atributos.
3. ♣ Defina uma classe Grafo, que suporte grafos de acordo com a seguinte interface:

Método - assinatura	Descrição
vertex_count()	Devolve o número de vértices do grafo
vertices()	Devolve um iterador sobre todos os vértices
edge_count()	Devolve o número de arestas do grafo
edges()	Devolve um gerador sobre todas as arestas do grafo
get_edge(u, v)	Devolve a aresta entre o vértice u e v , se existir
degree(v)	Devolve o grau do nó.
insert_vertex(v)	Atualiza o grafo com o vértice v
insert_edge(a)	Atualiza o grafo com a aresta a
remove_vertex(v)	Remove do grafo o vértice v e todas as suas arestas incidentes
remove_edge(e)	Remove a aresta e

O construtor do grafo deverá permitir a indicação do tipo de grafo, ou seja, se é dirigido ou Não dirigido.

- a) Sugere-se a implementação usando uma representação de mapas de adjacências, com a construção da sequência de vértices e dos "contentores" associados a cada vértice como dicionários Python em dois níveis: no primeiro nível, a chave é o objeto vértice e o valor o contentor. No contentor (o 2º nível), a chave é o vértice adjacente e o valor é a aresta.
- b) Sugere-se a implementação usando uma representação de listas de adjacências, com a construção da sequência de vértices e dos "contentores" associados a cada vértice como dicionários Python: a chave é o objeto vértice e o valor a lista de adjacências.

Exercício Extra ♣

No que se segue, vamos utilizar o conjunto de dados em [lisbon.lines.csv](#), que descreve as estações e localizações (geo-localizações) do metro de Lisboa.

1. Em Vertex, crie uma subclasse Station, para conter o id da estação e a posição, em graus decimais, da localização do objeto à superfície da terra.
2. Em Edge, crie uma subclasse Edge_line, para conter a linha de que a conexão faz parte.
3. Crie uma lista de objetos Station a partir do conjunto de dados em [lisbon.stations.csv](#) (estude o módulo csv do Python para facilitar a leitura e carregamento dos dados).
4. Crie uma lista de objetos "Edge_line" a partir do mesmo conjunto de dados.
5. A partir dos das listas dos exercícios 3.) e 4), crie um Grafo, usando a classe implementada no exercício 3.) da Parte 1
6. Crie uma função para poder visualizar a rede.

Exemplo:

```
import matplotlib.pyplot as plt
fig = plt.figure()
plt.line(x,y, c=cor) # para visualizar as conexões em que x e y são listas com as posições no
mapa das estações terminais de um Edge (executar tantas vezes quanto o número de
conexões).
```

Codifique a cor conforme apresentado na tabela em [lisbon.lines.csv](#).

Nota: Faça a projeção dos graus decimais indicados para o plano 2D, multiplicando a latitude por 1,26. Em alternativa, e para maior precisão, pode pesquisar fórmulas de cálculo de distâncias ortodrómicas a partir da latitude e da longitude.