

# Atv1\_eval2\_grafos

Siga as seguintes orientações:

- (o0) Faça parte de um grupo de dois integrantes (dupla).
- (o1) Estude o código `test_networkx_00.py`.
- (o2) Lembre-se que `V` é um dicionário com a estrutura indicada no código.
- (o3) Lembre-se que `E` é uma lista com a estrutura indicada no código.
- (o4) Lembre-se que `E_label = edge_label(E)`.
- (o5) Lembre-se que `path = dijkstra(G, 'S')`.
- (o6) Salve uma cópia de `test_network_00.py` com o nome `test_network_01.py`.
- (o7) Faça (c1), (c2), (c3) e (c4) no código `test_network_01.py`.
- (o8) Envie um e-mail para [jose.miguel@ci.ufpb.br](mailto:jose.miguel@ci.ufpb.br) com os nomes da dupla até a data da prova 2, anexando o arquivo `test_network_01.py`.

**(c1)** Altere a função `plot_graph` para que:

- Os pesos das arestas sejam mostrados verticalmente;
- As etiquetas dos vértices fiquem dentro de uma circunferência e a cor do interior desta seja branca;
- A função receba um argumento `edge_color`, de modo que as seguintes linhas funcionem:

```
plot_graph(G, V, E_label, edge_color='blue') # desenha as arestas com a cor azul
plot_graph(G, V, E_label, edge_color='red')  # desenha as arestas com a cor vermelha
plot_graph(G, V, E_label, edge_color='green') # desenha as arestas com a cor verde
```

**(c2)** Defina a função `P, VP, EP, EP_label = subgraph_from(path, V, E, E_label)` que constrói um subgrafo de  $(V, E)$  com arestas etiquetadas `E_label`, a partir da entrada `path`. As saídas `P, VP, EP, EP_label` correspondem a:

- `P`: subgrafo de  $(V, E)$  associado a `path`;
- `VP`: vértices de `path`;
- `EP`: arestas de `path`;

- `EP_label` : etiquetas de `EP`.

### Sugestões:

- Construa `VP` e `EP` a partir de `path`, empregando a mesma estrutura das variáveis `V` e `E`;
- Faça `EP_label = edge_label(EP)`;
- Faça `P = create_undirected_graph(VP, EP)`;
- Retorne `P, VP, EP, EP_label`.

**(c3)** Defina a função `plot_graphs(List_of_graphs)` para executar, sucessivamente:

```
plot_graph(G1, V1, E1_label, edge_color=ec1)
...
plot_graph(Gn, Vn, En_label, edge_color=ecn)
```

Onde `List_of_graphs = [[G1, V1, E1_label, ec1], ..., [Gn, Vn, En_label, ecn]]` e `n = len(List_of_graphs)`.

**(c4)** Tendo programado (c1), (c2) e (c3), adicione as seguintes linhas:

```
P, VP, EP, EP_label = subgraph_from(path, V, E, E_label)
plot_graphs([[G, V, E_label, 'blue'], [P, VP, EP_label, 'red']])
```