

Manual de utilização - task 2

Grupo constituinte

- Guilherme Gaspar n.º 2020218933
- Maria Carolina Fernandes n.º 2021218374
- Pedro Nuno Monteiro n.º 2021218544

Manual

Na pasta *Task 2* é possível encontrar 2 ficheiros *.py* (*functions.py* e *task2.py*) e uma pasta *networks* com 4 ficheiros de texto *.txt* que correspondem às 4 redes, obtidas da [SNDLib](#)

Ficheiro ***task2.py***

- Este é o ficheiro principal, aquele que deve ser corrido na linha de comandos para iniciar o programa
- Ao correr o programa, é apresentado um menu com 6 opções
 - Escolher 1 das 4 redes apresentadas
 - Inserir, em texto, o nome da rede
 - Sair
- Ao escolher uma das opções 1 a 4, o programa irá correr as 2 funções a seguir apresentadas, no ficheiro *functions.py*
- Escolhendo a opção 5, o programa irá apenas pedir ao utilizador para escrever, incluindo a extensão *.txt*, o nome da rede: para isso basta acrescentar na pasta onde estão presentes as restantes redes com a extensão *.txt*. Em seguida, basta escrever o nome do ficheiro adicionado, com a respetiva extensão.
- Ao escolher a opção 6, o programa é terminado

Ficheiro ***functions.py***

- Este ficheiro contém 6 funções que fazem de suporte ao programa principal, contendo as seguintes funções

1. *retrieve_data(data)*

- a. É a função que, ao receber a informação *data* (que corresponde ao conteúdo de um ficheiro *.txt*), vai buscar a informação relativa aos **nós** e aos **arcos** de cada rede.
- b. A função faz esta busca e adiciona diretamente no grafo *G* linha-a-linha, inicialmente para os nós e depois para os arcos
- c. Retorna o grafo ***G*** e a a tabela ***node_mapping*** serve como tabela de conversão, ou seja, atribui a cada nó, um número para que seja mais fácil para o utilizador escolher o nó de origem e nó de destino.

2. *draw_empty_network(G, node_mapping)*
 - a. Recebe como argumentos o grafo G e a tabela de conversão *node_mapping*
 - b. ANtes de ser pedido os nós origem e destino, é apresentado o grafo G, com os respectivos nós já numerados, de forma a que os utilizadores possam antes de escolher, observar a rede na sua totalidade.
3. *ask_origin_destiny(node_mapping)*
 - a. Esta função recebe como argumentos a tabela ***node_mapping***
 - b. É responsável por pedir ao utilizador os respectivos nós de **origem** e **destino**, certificando que os mesmos existem.
 - c. Retorna os nós origem e destino
4. *find_best_paths(G, origem, destino)*
 - a. Recebe como parâmetros o grafo G, o nó de origem e o nó destino
 - b. Calcula o caminho mais curto entre os nós de origem e destino, através do método Djisktra
 - c. Em seguida, copiamos o grafo original e removemos todos os nós que pertenciam ao primeiro caminho, e calculamos o segundo caminho mais curto
 - d. Este segundo caminho mais curto pode não existir se o nó for de primeira ordem ou se for um nó de
5. *draw_network(G, node_mapping, origem, destino, caminho1, caminho2)*
 - a. Esta função recebe como parâmetro o grafo G, previamente criado pela função anterior. Recebe também a tabela de conversão representada por *node_mapping*, o nó *destino* e o nó *origem*, e os caminhos previamente calculados pelo **Two Step Approach**.
 - b. É responsável pelo desenho do gráfico, utilizando funções do *Matlab*, apresentado os dois caminhos mais curtos, com cores diferentes.
6. *clear_screen()*
 - a. Apaga o ecrã