

Grupo 5

nº62372 Guilherme Alexandre Cardoso Soares

nº62211 Vitória de Mendonça Teixeira Correia

Funções extra implementadas e o seu funcionamento:

Safelevadas - (executa o programa) lê os dois arquivos de entrada contendo a rede de levadas e os pares de estações, entregando o *output* pretendido

devolveLista - módulo que inclui as funções responsáveis por encontrar os 3 caminhos mais curtos entre duas estações:

- **DFS** - foi aproveitado o código disponibilizado no manual da unidade curricular, adaptando-o para incluir o tempo em minutos e seguir a especificação pretendida para este projeto;
- **Search** - devolve os 3 caminhos mais curtos tendo em conta os nodes de partida e chegada fornecidos.

Edge - módulo contendo a classe dos objetos do tipo *Edge*:

- Foi acrescentado um atributo que representa os minutos associados ao *edge*;
- **getMins** - devolve o valor dos minutos;
- **__str__** - devolve string com informação sobre um determinado *edge*, incluindo minutos, para facilitar na testagem.

Node - módulo contendo a classe dos objetos do tipo *Node*:

- Foi apenas acrescentado um atributo com o id do *Node* relativamente ao código disponibilizado.

Digraph - módulo contendo a classe do objeto do tipo *Digraph*:

- Foi acrescentado um atributo que representa os minutos associados às ligações entre os *Nodes* (*edges*).
- **getEdgesMins** - devolve todos os valores dos minutos dos *edges*

fileProcessing - módulo com funções de leitura, extração e tratamento de informação dos ficheiros de *input*:

- **readLN** - extrai informação sobre a levada *network*;
- **readmyStations** - extrai informação sobre os pares de estações;
- **findSrcDestNodes** - encontra na *network* os nodes correspondentes às estações de partida e chegada;
- **buildNodes** - cria objetos do tipo *Node*;
- **buildEdges** - cria objetos do tipo *Edge*;
- **buildNetwork** - cria um directed graph com a informação extraída.

findAllStations - módulo que trata de encontrar os 3 caminhos mais curtos para cada par de estações:

- **findAllStations** - encontra os 3 caminhos mais curtos para cada par de estações, tratando também dos casos em que não é possível encontrar qualquer caminho

writeMS - módulo que lida com a escrita do ficheiro *output*:

- **writeMS** - escreve o ficheiro output com o resultado da procura dos 3 caminhos mais curtos entre cada par

TestSets:

Os testsets desenvolvidos usam um grafo comum a todos.

testset1 - testset mais elementar entre os 3. Testa apenas a capacidade do programa encontrar 3 caminhos mais curtos. Não inclui situações em que estações não comunicam ou não existem dentro da network.

testset2 - testset mais completo que o 1º. Para além da capacidade de encontrar os 3 caminhos mais curtos, testa casos de empate em termos de tempo e número de estações. Testa também situações de estações que não comunicam, ou que não existem dentro da network, no caso, da primeira estação não pertencer à network.

testset3 - Para além dos casos já testados nos testsets anteriores, testa casos em que existem menos de 3 caminhos possíveis. Testa empate em termos de somente tempo e também quando não existem dentro da network a segunda estação mencionada.

Divisão de tarefas:

Vitória de Mendonça Teixeira Correia nº62211

devolveLista.py: DFS(sort);

fileProcessing.py: readmyStations / findSrcDestNodes / isEdgeBi / findAllStations-
exceções “out of network” e “do not communicate”;

writeMS.py: exceções “out of network” e “do not communicate”

Node.py

Edge.py

testSets

Guilherme Alexandre Cardoso Soares nº62372

devolveLista.py: DFS(incorporação dos minutos) / Search

fileProcessing.py: readLN / buildNodes / buildEdges / buildNetwork

safelevadas.py

constants.py

findAllStations.py

writeMS.py

Digraph.py