Projeto de Programação II

Grupo: 3

André Patrício Alexandre Nº62224 Ana Rita Nicolau Nº58000

Rota do trabalho

Depois de lermos o relatório, averiguámos qual seria a estrutura básica de resolução do projeto. Analisámos o código base, de acordo com as especificações dadas pelo professor tanto nas aulas como no código em si. Reparamos que seria fundamental, numa instância inicial, criar no início um testSet para orientar o desenvolvimento, tendo em conta a verdadeira forma de resolução e as diferenças desta comparativamente com a resolução do problema anterior (código antes de qualquer alteração). A partir daí, criámos uma função que permitisse ler o testSet por nós criado. Nesta fase tínhamos já material suficiente para desenvolver de forma consistente uma possível resolução do projeto.

Analisámos o DFS e percebemos a forma como decidia os melhores caminhos. Aí tivemos que incluir um parâmetro "acumulate" que guardasse o tempo que cada caminho levava e, além disso, arranjar uma forma de decidir e guardar então os melhores caminhos. Tendo em conta as especificações dadas, criámos uma classe "Organiser" que possui métodos de seleção e organização dos melhores caminhos.

Trabalho

Organiser – André Alexandre & Ana Rita Nicolau

A Classe Organiser é responsável por várias tipologias de organização, estando essas divididas em duas vertentes: (1) organização geral e (2) organização/decisão no caso de possível adição de caminho. No caso da primeira, existe uma organização puramente baseada nas diretrizes de ordenação do projeto. Já na segunda, tendo em conta a solução não muito otimizada de adicionar todos os caminhos encontrados, acaba por decidir os caminhos entre si e guardar apenas os três melhores (caso existam três, claro).

safeLevadas – André Alexandre & Ana Rita Nicolau

Apesar de falarmos das funções infoFromFiles e infoToFiles, achamos que é de toda a pertinência falar sobre o modo de funcionamento do ficheiro safeLevadas tendo em conta as alterações que sofreu. Em geral, todas as pequenas alterações realizadas no projeto, alterações essas que não requeriram a criação de novas funções ou classes mas antes a renovação de já existentes, tiveram origem em alterações necessárias para o bom e correto funcionamento da função DFS. É à volta dela que todo o projeto se desenvolve e que são analisadas as informações provenientes dos ficheiros de input.

infoFromFiles – André Alexandre

A função infoFromFiles lê e reescreve as informações contidas nos ficheiros myStations.txt e myLevadasNetwork.txt. Ao analisar este último ficheiro referido, também verifica se existe alguma conexão omitida num dos seus sentidos e adiciona essa conexão.

infoToFiles – Ana Rita Nicolau

A função infoToFiles capta a lista de caminhos possíveis e analisa-os conforme a informação neles presente. Assim sendo, é responsável por escrever as informações no ficheiro myResults.txt.

testSets – André Alexandre & Ana Rita Nicolau

Criámos 8 testSets diferentes que analisam um leque de situações distintas e 4 grafos de levadas diferentes que servem de base a esses mesmos testSets. Alguns testSets partilham uma mesma levada sobre a qual testam coisas diferentes, e há grafos que possuem 2 redes que não se comunicam e outros que só possuem uma rede. Na pasta de cada testSet existe um documento de texto que explica qual a levada (grafo) usada e suas características, e o que é testado com ele.