

Programação 2 LTI, 2023/2024

Projeto 2

(este enunciado tem 12 páginas)

Levadas Seguras

Contexto

A progressiva concentração de cada vez mais pessoas e atividades nas cidades tem levado, por contraposição, a um cada vez maior número de amadores das caminhadas por ambientes e cenários naturais. Um dos destinos de eleição para caminhadas na natureza encontra-se em Portugal, na ilha da Madeira, mundialmente conhecida e apreciada pelos inúmeros trilhos que tiram partido das levadas de água e que permitem apreciar paisagens de extraordinária beleza.



Com o crescendo da procura e do número de praticantes, não raras vezes, esta atividade recreativa e saudável tem chegado ao conhecimento do grande público porém pelas piores razões, através de notícias de eventos dramáticos. Em muitos trechos, os percursos desenvolvem-se a grande altitude e são muito acidentados e perigosos. É com alguma frequência que nos chegam notícias de buscas encetadas pelas autoridades por caminhantes que não regressaram aos seus alojamentos e que se encontram perdidos ou isolados devido ao agravamento das condições meteorológicas. Infelizmente, mais vezes do que seria desejável, as notícias dão também conta de fatalidades.



Para aumentar o nível de apoio aos caminhantes, em geral, e de auxílio aos acidentados, em particular, as autoridades ponderaram constituir equipas de apoio e salvamento. Estas equipas teriam por missão irem percorrendo os diferentes trilhos, num esquema de revezamento ao longo do dia que permitiria garantir este serviço em permanência.

Acontece porém que se depararam com um obstáculo de monta para a viabilidade e eficácia desse programa. No seu conjunto, os trilhos formam uma rede de caminhos com uma extensão estimada em cerca de 2000 quilómetros. Por conseguinte é impraticável ter uma equipa em cada troço de cada trilho.



Acontece que estando as autoridades neste impasse prático, havia felizmente entre a equipa de gestão das levadas, um estagiário que era aluno de Programação 2 LTI e que de imediato se apercebeu que aquilo que estava a aprender sobre otimização poderia ser aplicado na mitigação deste problema. O que era preciso era desenvolver um programa que perante a chamada de auxílio de um caminhante, geolocalizada, e os pontos, também geolocalizados, em que se encontrassem cada uma das equipas de socorro, determinasse aquela equipa que se encontrava a menor tempo de caminho do ponto de chamada e que seria a que devia acudir à chamada.

Com facilidade convenceu os coordenadores dos serviços de apoio aos caminhantes para que estes financiassem a implementação de um sistema deste género. Como primeiro passo, e para apoiar fases subsequentes de desenvolvimento do sistema, precisava antes de mais de, dados dois pontos, algures na rede de levadas, determinar qual o percurso entre eles que requer menor tempo a ser percorrido. Para tanto pediu ajuda aos seus colegas de Programação 2, no que deu origem ao presente enunciado.

1. Aplicação a desenvolver

Objetivo

Com uma finalidade pedagógica, usando Python 3, neste projeto vai implementar o software `safeLevadas`. É um software que permite obter o percurso entre dois pontos de uma rede que demora menos tempo a percorrer.

No contexto das atividades da nossa disciplina, uma outra finalidade pedagógica encontra-se no desafio de se desenvolver software a partir de uma especificação muito mais difusa e aberta do que aquelas que foram recebidas como pontos de partida em projetos anteriores pelos alunos da disciplina.

Ainda outra finalidade pedagógica, consiste em os alunos ganharem prática em reutilizarem e adaptarem código pré-existente. Na resolução deste projeto, **é obrigatório recorrer a e adaptar o código disponibilizado no manual desta unidade curricular para a tarefa de otimização. Resoluções que recorram a outros módulos e/ou respetivas funções para realizarem esta tarefa, como por exemplo, scikit-learn, SciPy, Pyclustering, entre possivelmente outros, recebem zero valores de classificação.**

Funcionalidade

O seu programa recebe uma rede de levadas estabelecidas entre estações de referência. Recebe ainda duas determinadas estações A e B dessa rede.

O seu programa entrega **os três percursos (sequência de estações) que demoram menos tempo a percorrer** (em minutos) entre essas estações A e B de entrada.

Entrada

O programa recebe pares de nomes de estações e um ficheiro contendo a descrição da rede de estações, com uma estrutura interna para arrumação de informação similar à do seguinte exemplo fragmentário:

`myLevadasNetwork.txt`

```
#Id, Name, Connected:
A, Seixal, [(R, 15), (M, 8), (B, 12)]
C, Pico Ruivo, [(GJ, 32), (I, 5)]
D, Queimadas, [(Z, 18), (AC, 11)]
E, Ponta do Pargo, [(DW, 13)]
...
```

em que os campos `id` e `Name` de uma linha indicam o identificador e o nome de uma estação; e o campo `Connected` de uma linha guarda os pares com o `id` da estação com levada direta para essa estação descrita nessa linha e como o tempo necessário para a alcançar, em minutos.

Saída

O programa produz:

- "X out of the network" caso X não faça parte da rede;
- ou "X and Y do not communicate" caso não exista percurso entre X e Y;
- ou um par com um `int` que indica o tempo mais breve, em minutos, que se leva a chegar da primeira dessas estações à outra, e com a sequência de estações nesse percurso.

Especificação geral

- Um percurso existe entre estações ligadas diretamente por uma levada e indireta ou transitivamente entre estações ligadas por levadas que passam por outras estações.
- Entre duas estações X e Y ligadas indiretamente através da intermediação de outras estações diferentes de X e de Y, Y é alcançado a partir de X em N minutos, em que N é a soma dos tempos de cada ligação entre as diversas outras estações intermédias.
- Entre duas estações X e Y ligadas indiretamente, pode acontecer haver mais de um caminho através de outras estações que liga X e Y: pretende-se considerar o caminho a que corresponde o tempo mais breve possível para se alcançar Y a partir de X.
- Entre duas estações X e Y, ligadas diretamente por uma levada, Y é alcançável a partir de X em N minutos, em que N é indicado na linha do ficheiro `myLevadasNetwork.txt` que caracteriza X (ou Y).
- As relações diretas entre estações da rede de levadas são simétricas, ou seja se Y está em relação direta com X, de igual modo X está em relação direta com Y (mesmo que apenas uma destas duas direções esteja registada no ficheiro com a descrição da rede).
- Cada uma das estações tem de ser declarada no ficheiro `myLevadasNetwork.txt`, e é declarada apenas uma vez.

Estrutura da aplicação

Em vista de conter o projeto dentro dos seus limites pedagógicos, as pré-condições sobre a estrutura interna do ficheiro de input, respeitante ao formato de arrumação da informação, não devem ser verificadas (seguindo uma abordagem de programação por contrato, toma-se por garantido que o cliente da aplicação assegura que os ficheiros vêm bem estruturados).

Reutilização de código alheio

Os alunos têm reutilizar o código relevante constante do Capítulo 14 do manual, e disponibilizado na página moodle desta disciplina. O primeiro passo no desenvolvimento da resolução deste projeto consiste em copiar esse código. O trabalho subsequente de desenvolvimento consiste em ajustá-lo e alargá-lo no que for conveniente e necessário para se obter uma resolução. No âmbito deste projeto, esta cópia e reutilização de código alheio não é considerada violação da integridade

académica. Cópias e/ou reutilização de outros códigos alheios continuam a ser considerados violação da integridade académica.

Linguagem

A linguagem do input e output do software para utilizadores humanos é o inglês.

A linguagem veicular da documentação, especificação, nomeação de funções, variáveis, comentários no código etc é também o inglês.

Executar o software

O software é executado através da seguinte instrução na linha de comandos:

```
python safeLevadas.py inputFile1.txt inputFile2.txt outputfile.txt
```

em que:

- `inputFile1.txt` é um ficheiro com a rede de levadas num formato como exemplificado acima;
- `inputFile2.txt` é um ficheiro com um par de nomes de estações por linha, como no exemplo fragmentário abaixo:

`myStations.txt`

```
Cedro - Queimada
Ponta do Pargo - Calheta
Queimada - Boavista
Areeiro - Pico Ruivo
Moinho - Rabacal
...
```

- e `outputfile.txt` é o nome do ficheiro em que são escritos os resultados para cada par presente no `inputFile2.txt`, pela ordem respetiva do `inputFile2.txt`, como no exemplo fragmentário abaixo, em que o primeiro elemento é o tempo e os seguintes as estações do percurso:

`myResults.txt`

```
# Cedro - Queimada
76, Cedro, Queimada
83, Cedro, Rabacal, Queimada
120, Cedro, Ponta do Sol, Areeiro, Queimada
# Ponta do Pargo - Calheta
Calheta out of the network
# Queimada - Boavista
89, Queimada, Ponta do Sol, Pico Ruivo, Boavista
# Areeiro - Pico Ruivo
Areeiro and Pico Ruivo do not communicate
# Moinho - Rabacal
```

56, Moinho, Areeiro, Rabacal
56, Moinho, Popias, Rabacal
98, Moinho, Rabacal
...

Para cada pedido, as linhas dos respetivos três percursos são ordenados de cima para baixo por ordem crescente de duração. Em caso de empate no tempo, do que passa por mais estações para o que passa por menos estações. Em caso de empate ainda, em termos de número de estações, por ordem lexicográfica dos nomes da segunda estação no percurso.

No caso de haver menos de três percursos possíveis, indica-se aqueles que existirem abaixo desse número.

Conjunto de testes

Para exercitar os princípios e métodos ensinados em Programação 1 sobre testar e depurar, os estudantes têm de construir o seu próprio conjunto de testes e usá-lo para testar e depurar a resolução que desenvolverem. Os ficheiros com esses testes utilizados tem de ser submetido para classificação juntamente com os demais ficheiros de código desenvolvido pelos estudantes.

De modo diferente do que aconteceu nos projetos anteriores, o conjunto de testes a ser usado pelos docentes na classificação das resoluções deste projeto pelos estudantes não serão por isso divulgados antes do prazo para a submissão das resoluções.

2. Desenvolvimento do software

Grupos

O projeto tem de ser realizado por grupos de exatamente 2 estudantes. Os grupos não podem ser formados por elementos que já formaram grupo no projeto 1 de Programação 2 nem no projeto de Programação 1. Cada estudante ERASMUS deve fazer grupo com um estudante não-ERASMUS. Os grupos podem conter estudantes de diferentes turmas. Os grupos registam-se no site da disciplina.

A única FORMA DE REGISTO de grupos é através do site moodle da disciplina.

Elementos fornecidos aos alunos

Para a elaboração da componente de avaliação respeitante ao projeto, é fornecido o seguinte, que se encontra no site da disciplina:

- presente enunciado, que inclui especificação geral da aplicação
- cópia do código do Capítulo 14 a reutilizar obrigatoriamente (no respetivo sumário no moodle)

Máximas

Os estudantes a realizar o presente projeto são tipicamente programadores principiantes. Têm toda a vantagem em observar as seguintes máximas, que ainda não tiveram oportunidade de consolidar por si próprios:

1. "já"

positivo: começar a resolver o projeto agora, no momento em que este enunciado foi publicado

negativo: esperar até alguns dias antes do prazo de entrega para começar leva ao desastre

2. "passo a passo"

positivo: ir fazendo e testando pequenas partes do código progressivamente

negativo: esperar para testar até haver uma primeira versão total ou completa leva ao desastre

3. "desbloquear rápido"

positivo: falar com os docentes (e colegas) para esclarecer dúvidas e desbloquear impasses logo que estes surgem

negativo: esperar por futuro rasgo solitário de inspiração súbita leva ao desastre

Apoio para a resolução do projeto

Continuam ao dispor os meios de apoio pedagógico para os alunos desta disciplina, que se encontram disponíveis desde o início do curso, e que podem e devem ser usados para apoio à resolução do presente projeto. Relembra-se que são os seguintes:

- contato com os docentes ao **final das aulas** ao longo do semestre
- horários de **atendimento** presencial, individual e personalizado, aos alunos ao longo da semana (indicado no Guião da disciplina, na respetiva página moodle)
- **fórum de entreajuda** da disciplina, com acesso por todos os estudantes (na página moodle da disciplina)
- espaço de **notícias** da disciplina (na página moodle da disciplina)

Dada a natureza da tarefa a concretizar e o contexto do código em que eventuais dificuldades surgem, esclarecimentos sobre a resolução do projeto devem ser obtidos através destes meios de apoio, não sendo atendíveis através de mensagens de email para os docentes.

3. A componente de avaliação

Elementos a entregar pelos estudantes para avaliação

A entregar: uma pasta com o ficheiro com o **relatório** de implementação, com os ficheiros de **código** desenvolvidos e com os ficheiros de **teste** usados.

A pasta deve ter o nome `safeLevadasGroupN`, em que `N` é o número do grupo, atribuído no processo de inscrição do grupo. Por exemplo, para o grupo de estudantes que recebeu o número 249, a pasta deve ter o nome `safeLevadasGroup249`. A pasta tem de ser submetida zipada, com o nome `safeLevadasGroup249.zip`.

Cada um dos ficheiros de código, por sua vez, tem de conter nas primeiras linhas, como comentários, informação sobre o número do grupo e número e nome completo de cada membro do grupo que trabalhou no projeto, como exemplificado a seguir:

```
# 2023-2024 Programacao 2 LTI
# Grupo 249
# 92123 Felisberto Felismino
# 92456 Cátia Vanessa
```

Ficheiros sem algum destes elementos não serão avaliados.

Relatório de implementação

O relatório de implementação, com um **máximo de duas páginas**, tem o nome `relGroupN.pdf` (em que `N` é o número do grupo) e tem de estar no formato `.pdf` (relatórios noutros formatos serão ignorados). Tem de ser estruturado de acordo com as seguintes **secções**:

1. Número do grupo
2. Número e nome completo de cada membro do grupo
3. Indicação detalhada do que cada membro do grupo fez para a resolução do projeto
4. Indicação de funções extra implementadas e do seu funcionamento
5. Indicação das funcionalidades que ficaram por implementar (se aplicável)
6. Indicação de erros conhecidos (se aplicável)

O relatório pode ser escrito em português ou em inglês.

Dimensões em avaliação

Os projetos serão avaliados de acordo com as seguintes dimensões e ponderações:

- A. 1 se está completo e funciona sem gerar erros ao compilar e correr sobre os testes usados pelos docentes para avaliação, 0 caso contrário
- B. Correção semântica (funciona como especificado no enunciado), 50%
- C. Conjunto de testes (abrangentes, sucintos, etc), 10%
- D. Correção pragmática (reutilização do código do Cap. 14, estruturação, factorização, etc), 20%
- E. Documentação (especificação, comentários q.b.), 10%
- F. Legibilidade (apresentação, arrumação e formatação do código), 5%
- G. Relatório de implementação, 5%

A classificação é encontrada através da fórmula $A * (B + C + D + E + F + G)$

Integridade académica

A resolução submetida para avaliação tem de ter sido concebida, codificada, testada e documentada na íntegra apenas pelos seus autores, que são os dois membros do grupo, [sem prejuízo de que esses elementos do grupo têm de reutilizar, adaptando, o código do Caps. 14 \(2ªed.: 12\) do manual usado na disciplina.](#)

Alunos detetados em situação de fraude, incluindo a aquisição a terceiros ou plágio parcial ou total - plagiadores e plagiados, com ou sem a intervenção de intermediários - em alguma componente de avaliação ficam liminarmente com esta prova cancelada e serão alvo de processo disciplinar, o que levará a um registo dessa incidência no processo de aluno. Não queira ter de mostrar o seu diploma a um futuro empregador com uma incidência dessas registada.

Pode e deve haver entreajuda entre alunos, através da discussão de abordagens e algoritmos aplicáveis. É porém da exclusiva responsabilidade de cada grupo tomar medidas para proteger o seu código de ser plagiado.

No processo de avaliação será usado software de apoio na deteção de plágio que compara a resolução de cada grupo com cada uma das resoluções dos outros grupos. Como esperado para a sua funcionalidade, o software deteta tentativas de ludibriação, como, por exemplo, renomeação de variáveis, de funções, de constantes, etc, reordenação das funções no código, inserção de linhas em branco extra, etc, etc. Evitem espertezas parvas, para evitarem desastres amargos.

Forma e data de entrega

Para submeterem a solução do vosso grupo a avaliação, **entregam um FICHEIRO .zip**, que resulta de se comprimir a pasta com os ficheiros de código desenvolvidos, de teste e o relatório (por exemplo, safeLevadasGroup249.zip).

A **ÚNICA FORMA DE ENTREGA é a através do site da disciplina no moodle**

Qualquer entrega noutra forma não será considerada para avaliação.

Para ser avaliada, a vossa solução deve ser submetida até ao **PRAZO de sexta-feira, 31 de maio de 2024, 23h00 (hora de Lisboa).**

Qualquer entrega ou resubmissão depois deste prazo não será considerada para avaliação.