



## Processamento de um mapa

A NASA está a preparar uma nova missão para a sonda espacial Curiosity que se encontra em Marte. Para planear essa missão vai utilizar um mapa obtido pelo satélite Dawn durante o seu caminho para os asteróides de Marte.

O seu projecto vai processar o mapa obtido pelo satélite Dawn com dados sobre uma certa região de Marte. O satélite Dawn não consegue enviar o mapa na sua totalidade, mas é capaz de responder a questões sobre a informação contida no mapa.

Sabe-se que o mapa corresponde a uma matriz com um certo número de linhas e um certo número de colunas. Em cada posição desta matriz, dada pela especificação de uma coluna e de uma linha, está armazenada a informação referente a um ponto existente no mapa.

Os engenheiros da NASA criaram um protocolo de comunicação com o satélite Dawn para obter informação sobre o mapa. Este protocolo utiliza duas funções que comunicam com o satélite, o qual responde fornecendo informação relevante sobre o mapa:

- `dimensoes()`

Esta função origina uma comunicação com o satélite, o qual devolve um tuplo com dois elementos, no qual o primeiro elemento contém o número de colunas existentes no mapa e o segundo elemento contém o número de linhas existentes no mapa.

- `conteudo(c, l)`

Esta função origina uma comunicação com o satélite, o qual devolve a informação contida na posição do mapa correspondente à coluna  $c$  e à linha  $l$ . A origem do mapa corresponde à posição  $(0, 0)$ , que se encontra no canto inferior esquerdo do mapa (ver a Figura 1). Esta informação pode conter o seguinte:

- Um número que indica a temperatura medida no ponto do mapa;
- O carácter ' ' que indica que devido a ruído existente na comunicação, a informação obtida sobre esse ponto no mapa não é conhecida.

## 1 Trabalho a realizar

O seu programa, usando o protocolo de comunicação da NASA, deverá calcular a temperatura média num caminho entre dois pontos do mapa. O ponto de início deste caminho

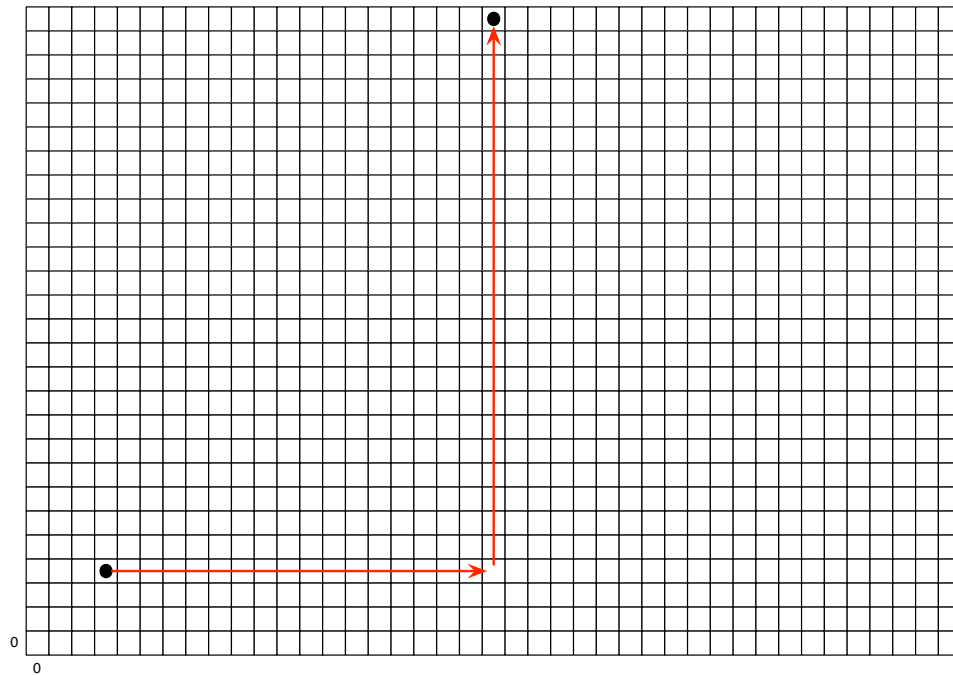


Figura 1: Exemplo de um caminho.

é fornecido ao seu programa através de uma posição especificada por uma coluna e por uma linha no mapa.

As funções correspondentes ao protocolo de comunicação com o satélite Dawn encontram-se no ficheiro `NASA.pyc` disponível na página da disciplina. Este ficheiro deverá ser colocado na directoria do seu computador onde o ambiente de desenvolvimento cria por omissão os ficheiros com extensão `py`. O seu programa deverá conter a instrução `from NASA import *`, a qual torna disponíveis as funções `dimensoes` e `conteudo`.

A partir do ponto de início, o caminho percorrido corresponde a seguir a mesma linha do mapa até ser atingido o ponto médio do mapa (se o mapa tiver um número par de linhas ou de colunas, o ponto médio corresponde à divisão inteira para o cálculo do ponto médio). A partir do ponto médio, o caminho vai seguir essa coluna do mapa, deslocando-se ou para cima ou para baixo, seguindo o caminho mais longo até à fronteira do mapa (se a linha corresponder ao ponto médio na vertical, o seu programa desloca-se para cima).

Ao encontrar um ponto para o qual não existe informação, o seu programa regista a posição desse ponto e calcula a temperatura como sendo a média das temperaturas nos oito pontos do mapa que circundam esse ponto. Se algum destes pontos também não conter informação, esse ponto é ignorado no cálculo da média. No caso extremo de nenhum dos pontos que circundam um dado ponto contiver informação, então utiliza-se o último valor registado para a temperatura ao longo do caminho. No seu programa deve tomar duas decisões: (1) o que fazer se o ponto para o qual não existe informação corresponder a uma fronteira do mapa (ou seja, certos pontos que o circundam não existem no mapa) e (2) o que fazer se não tiver sido registado nenhum valor anterior para a temperatura.

O seu programa deverá produzir informação sobre o caminho seguido, através da indicação do ponto de partida, do ponto no meio do mapa em que a direcção foi mudada e o

ponto no final do mapa. Deverá indicar qual o valor médio da temperatura ao longo do caminho percorrido e quais os pontos para os quais não existia informação e a temperatura calculada para esses pontos.

O seu programa deve corresponder a uma função com o nome `temperaturas`, a qual recebe dois argumentos inteiros correspondentes, respectivamente, à coluna e à linha da posição inicial. O seu programa devolve uma lista com os seguintes elementos:

1. Um tuplo com a coluna e a linha da posição inicial;
2. Um tuplo com a coluna e a linha do ponto em que se verificou uma mudança de direcção;
3. Um tuplo com a coluna e a linha da posição final no mapa;
4. Um número que corresponde ao valor médio da temperatura ao longo do caminho;
5. Uma lista de tuplos, em que cada tuplo contém como primeiro elemento um tuplo correspondente a uma posição encontrada sem valor registado (representado por uma coluna e uma linha) e o segundo elemento contém um número que corresponde ao valor da temperatura calculada para esse ponto.

O seguinte exemplo mostra uma utilização do programa, no qual “...” omite muitos dos valores devolvidos:

```
temperaturas(345, 790)
[(345, 790), (1400, 790), (1400, 1300), -19.88005050505051,
 [((347, 790), -10.0), ((349, 790), -10.0), ...]]
```

## 2 Sugestões

1. Comece por desenvolver uma função que calcula o valor da temperatura num ponto do mapa. A sua função deve tratar tanto dos casos em que a temperatura está explicitamente registada como dos casos em que a média das temperaturas dos pontos circundantes tem que ser calculada. A sua função poderá também devolver informação adicional que é necessária para o programa. Teste a sua função cuidadosamente, de modo a garantir que esta funciona em todos os casos possíveis.
2. Escreva uma função que calcula o caminho a percorrer no mapa a partir de um ponto inicial, devolvendo o ponto inicial, o ponto em que se muda de direcção e o ponto final do caminho. Teste a sua função cuidadosamente, de modo a garantir que esta funciona em todos os casos possíveis.
3. Desenvolva o resto do seu programa utilizando as duas funções que já escreveu e que já sabe que funcionam correctamente.

### 3 Aspectos a evitar

Os seguintes aspectos correspondem a sugestões para evitar maus hábitos de trabalho (e, consequentemente, más notas no projecto):

1. Não pense que o projecto se pode fazer nos últimos dias. Se apenas iniciar o seu trabalho neste período irá ver a Lei de Murphy mesmo em funcionamento (todos os problemas são mais difíceis do que parecem; tudo demora mais tempo do que nós pensamos; e se alguma coisa puder correr mal, ela vai correr mal, na pior das alturas possíveis).
2. Não pense que um dos elementos do seu grupo fará o trabalho por todos. Este é um trabalho de grupo e deverá ser feito em estreita colaboração (comunicação e controle) entre os vários elementos do grupo, cada um dos quais com as suas responsabilidades. Tanto uma possível oral como perguntas no segundo teste sobre o projecto, servem para despistar estas situações.
3. Não duplique código. Se duas funções são muito semelhantes é natural que estas possam ser fundidas numa única, eventualmente com mais argumentos.
4. Não se esqueça que as funções com muitas linhas são penalizadas no que respeita ao estilo de programação.
5. A atitude “vou pôr agora o programa a correr de qualquer maneira e depois preocupo-me com o estilo” é totalmente errada.
6. Quando o programa gerar um erro, preocupe-se em descobrir *qual* a causa do erro. As “marteladas” no código têm o efeito de distorcer cada vez mais o código.

### 4 Classificação

A nota do projecto será baseada nos seguintes aspectos:

1. Execução correcta (70%).  
Esta parte da avaliação é feita recorrendo a um programa de avaliação automática que sugere uma nota face aos vários aspectos considerados.
2. Facilidade de leitura, nomeadamente abstracção procedimental, nomes bem escolhidos, paragrafação correcta, qualidade (e não quantidade) dos comentários e tamanho das funções (25%). Para além de outra informação relevante, os seus comentários devem indicar de forma clara quais as decisões tomadas no cálculo da temperatura de um dado ponto (ver a Secção 1).
3. Estilo de programação (5%).

### 5 Condições de realização e prazos

A primeira parte do projecto deve ser entregue até às **15:00** horas do dia **14 de Novembro de 2012**, na sala de estudo do DEI ou na portaria do Tagus (consoante o campus que

corresponda ao grupo do projecto), e deverá constar de uma listagem do código em fonte monospace. A listagem deve ser entregue agraphada e não encadernada nem dentro de uma capa.

Projectos em atraso serão aceites até ao dia 16 de Novembro às 15:00 horas, sendo penalizados com **1 valor por cada dia de atraso**. A partir das **15:00** horas do dia **16 de Novembro de 2012** não se aceitam quaisquer projectos, seja qual for o pretexto.

Para além disto, a submissão do código por via electrónica, através do sistema Fénix, é obrigatória e deverá ser feita nos mesmos prazos que a entrega do relatório.

O código do projecto deve estar contido num único ficheiro com o nome `FP1213-parce1-grupo<n>.py`, em que `<n>` é o número do seu grupo. Por exemplo, o ficheiro do grupo número 5 deverá chamar-se `FP1213-parce1-grupo5.py`. O ficheiro de código deve conter em comentário, na primeira linha, os números e os nomes dos alunos do grupo, bem como o número do grupo.

Pode ou não haver uma discussão oral do trabalho e/ou uma demonstração do funcionamento do programa (será decidido caso a caso).

Projectos iguais, ou muito semelhantes, serão penalizados com a reprovação na disciplina. O corpo docente da cadeira será o único juiz do que se considera ou não copiar num projecto.