

Truques de programação IV - Python

Nesta folha você deve resolver alguns exercícios de programação. Cada um deles sugere um truque que quando apreendido pode ser usado em inúmeros outros problemas parecidos ou não.

Para seu processamento, você deve ler o arquivo

F186001.myd

publicado no lugar usual.

Matrizes simétricas Escreva um programa que analise uma matriz e informe se ela é ou não é simétrica. Para esta verificação, os elementos simétricos em relação à diagonal principal devem ser iguais. A matriz precisa ser quadrada e nela todos os elementos $m[i][j]$ precisam ser iguais a $m[j][i]$. Por abuso, esta regra pode ser aplicada aos elementos que estão na diagonal principal, já que quem está nela tem $i = j$ e portanto $m[i][j]$ sempre será igual a $m[j][i]$ quando $i = j$. Mas, nada se perde se elementos da diagonal principal simplesmente não forem testados. De acordo com a regra, basta que um único elemento simétrico seja diferente para que a matriz toda o seja. Um possível gabarito:

```
import numpy as np
def f186():
    f=open("c:/p/n/186/f186001_exemplo.myd","r")
    i=ct=0
    m=np.zeros((10,10))
    while i<10:
        for ii in range(10):
            x=f.readline()
            nums=x.split()
            for jj in range(10):
                m[ii,jj]=int(nums[jj])
        sentinela=0
        for ii in range(10):
            for jj in range(10):
                if m[ii,jj]!=m[jj,ii]:
                    sentinela=1
        if sentinela==0:
            ct=ct+1
        i=i+1
    print("Simetricas = ",ct)
```

O truque: Sentinela é uma variável que sinaliza uma determinada condição ao longo de um processamento completo. Usada muitas vezes, como neste contexto, no qual uma única condição perdida em um volume grande de situações e lugares (como um único elemento simétrico diferente) pode determinar uma condição mais geral (toda a matriz deixou de ser simétrica).

Para otimizar o processamento, ao invés de usar os dois ciclos *for* poderíamos ter usado ciclos *while*, incluindo neles (com AND) a condição *sentinela* = 0, para que, uma vez descoberto um elemento diferente a busca fosse encerrada. **Para você fazer** Leia no arquivo acima descrito, 10 matrizes numéricas de 10 linhas por 10 colunas estão registradas. O objetivo do aluno deve ser contar quantas são simétricas.

matrizes simétricas:

Vulcão no Havai Como sabemos o Havai está neste momento em processo de manifestação vulcânica intensa. O vulcão Saidebaixo, numa ilha lá, tem algumas características interessantes: a) fica numa ilha que tem platôs quadrados, de altura regular e é deserta. b) sua lava é muito líqüida, quase líquida, saindo da cratera e escorrendo, se possível, para as 8 direções vizinhas. c) não há explosões, a lava sai e escorre apenas pela ação da gravidade quando possível. d) a cada erupção, a cratera aparece em um local

distinto da ilha. Você, como estagiário de geologia, recebeu a incumbência de escrever um programa que: a) leia uma matriz NxN, que representa as alturas médias dos platôs da ilha (cada platô tem 1 hectometro², 1 hm=100m) em hm, em relação ao nível do mar. Assim, se na posição 2,2 houver o valor 3, isto indicará que neste hectômetro quadrado a altura média é de 300m em relação ao mar. b) leia uma posição x,y representando a localização da cratera, nesta erupção. Essa localização já está na modalidade desejada, ou seja, variando entre 0 e N-1. Especial cuidado deve ser tomado para o caso em que a cratera já esteja na margem da ilha. Neste caso ela poderá não ter alguns dos 8 vizinhos.

Sabe-se que a lava flui apenas do platô cratera para os platôs vizinhos se estes tiverem uma quota menor (forem menos altos), e que a lava vaza para os 8 vizinhos (nas direções N, NE, E, SE, S, SO, O e NO) à velocidade de 1 platô/hora.

Quando a lava chega nos limites da ilha, ela cai no mar e deixa de ter relevância para este problema.

Supondo que a erupção ocorre às 14hr, seu programa deve calcular e imprimir quantos hectômetros quadrados foram cobertos de lava após 1 hora da erupção, ou seja as 15h00. Deve imprimir também qual a altura mínima da lava nesta hora, em relação ao nível do mar. Note que a quantidade de platôs da resposta deve variar entre 1 (só a cratera) e 9 (A cratera e seus 8 vizinhos).

Para seu entendimento, suponha uma ilha com a configuração

1 1 7 8 2 2	Aqui:
2 2 3 4 1 9	crat=4.0 => p=1 m=3
3 3 5 6 7	2,2 5 2
4 5 6 7 8 2	3,3 8 4
3 3 4 5 6 7	5,5 1 1
3 4 5 2 2 1	

```
i=0
platos=sommin=0
while i<20:
    for ii in range(10):
        x=f.readline()
        nums=x.split()
        for jj in range(10):
            m[ii,jj]=int(nums[jj])
    x=f.readline()
    nums=x.split()
    xx=int(nums[0])-1
    yy=int(nums[1])-1
    qtd=0
    minimo+=999999
    for ii in range(xx-1,xx+2):
        for jj in range(yy-1,yy+2):
            if ii>0 and ii<10 and jj>0 and jj<10:
                if m[ii,jj]<m[xx,yy]:
                    qtd=qtd+1
            if (ii!=xx or jj!=yy) and m[ii,jj]<minimo:
                minimo=m[ii,jj]
    platos=platos+qtd
    sommin=sommin+minimo
    i=i+1
print("Vulcao = ",platos," Minimos = ",sommin)
```

O truque: Achar os vizinhos de um determinado elemento em uma matriz. Basta subtrair e acrescentar 1 unidade (ou mais, se forem vizinhos mais distantes) nos índices dos elementos a processar. Especial cuidado deve ser tomado nos limites da matriz. O número de vizinhos só é total nos elementos centrais da matriz, ou seja, aqueles que não estão em nenhuma borda.

Para você fazer: No arquivo acima, há 20 vulcões de 10 x 10, abaixo de cada um há uma coordenada da cratera. Você deve processar os 20 vulcões e para cada um deve-se achar a soma dos platos atingidos e a soma das quotas mínimas em cada vulcão.

$\sum$ platos	$\sum$ quotas _{min}
---------------	------------------------------

primeiros termos valem 1 e 1 e a partir do terceiro, cada elemento é igual à soma dos dois anteriores. Neste exercício, você está sendo convidado a desenvolver um gerador de seqüências de Fibonacci genérico. A diferença é que os 2 primeiros termos da seqüência serão variáveis, embora a regra de construção dos elementos além do segundo permaneça a mesma, isto é, a soma dos dois anteriores. Eis alguns exemplos destas seqüências genéricas:

2,7,9,16,25,41,...
3,7,10,17,27,44,71,115,...
100,200,300,500,800,1300,2100,3400,...
123,456,579,1035,1614,2649,4263,...

Você deve escrever um programa e nele deve escrever uma função de nome fibonacci(), que receba 3 parâmetros: a saber (a,b,q) a é o primeiro termo, b o segundo termo, e q é a ordem do elemento que deve ser calculado. Por exemplo, se q=3 então é o terceiro a ser calculado, se q=7 é o sétimo e em termos genéricos é o q-ésimo termo.

O programa principal deve receber uma quantidade indeterminada de triplas a,b,q e para cada uma calcular os termos de fibonacci pedidos através do parâmetro q.

Note que além de devolver o q-ésimo termo, a função retornar também o termo anterior (o de ordem q-1), já que ambos deverão ser impressos.

```
Exemplos de uso:
Informe a, b e q: 1 1 7
Termo q-1: 8 - Termo q: 13
Informe a, b e q: 2 7 6
Termo q-1: 25 - Termo q: 41
Informe a, b e q: 3 7 7
Termo q-1: 44 - Termo q: 71
Informe a, b e q: 123 456 5
Termo q-1: 1035 - Termo q: 1614
Informe a, b e q: 0,0,0
```

```
def fibo(a,b,qual):
    i=3
    while i<=qual:
        res=a+b
        a=b
        b=res
        i=i+1
    return [a,b]
```

```
...
i=somq=somqm=0
while i<200:
    x=f.readline()
    nums=x.split()
    a=int(nums[0])
    b=int(nums[1])
    qual=int(nums[2])
    zz=fibo(a,b,qual)
    somq=somq+zz[1]
    somqm=somqm+zz[0]
    i=i+1
    print(somqm,somq)
```

No arquivo acima, há 200 triplas a,b,q, devendo-se calcular os termos X_{q-1} e X_q e somar os 200 resultados.

$\sum X_{q-1}$	$\sum X_q$
----------------	------------

Para você fazer

simétricas	$\sum$ platos	$\sum$ min
X_{q-1}	X_q	////// ////// ////// ////// ////// ////// ////// //////



Sequencia de Fibonacci Os historiadores da matemática quase garantem que a disseminação dos algarismos arábicos (e também do zero, este de origem hindu) na cultura ocidental, vem do livro de Fibonacci, Liber Abacci publicado na Itália por volta de 1220.

Na nossa matemática contemporânea, o Fibonacci está associado à célebre seqüência de números: 1,1,2,3,5,8,13,21,34,55,... na qual os dois

===== 04/12/2019 10:49:28.2 =====E=PL186p
1 3 76 63 1111828 1798904