

Truques de programação VI - Python

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

Para seu processamento, você deve ler o arquivo

F188001.myd

publicado no lugar usual.

Ponto central Este problema surge em diversos contextos (ainda mais nos dias de hoje, com a popularização da geolocalização e georeferência via GPS). Trata-se de uma coleção arbitrária de pontos, devendo-se escolher o ponto central do conjunto. A definição formal sugere escolher o ponto que tiver o menor maior raio é o centro. **Truque:** Usa-se aqui o Teorema de Pitágoras para descobrir distâncias, já que estas sempre ou quase sempre virão na forma de coordenadas x, y o que sugere imediatamente um triângulo retângulo.

```
def f188():
    f=open("c:/p/n/188/f188001_ex2.myd","r")
    ctd=0
    i=0
    maximo=np.zeros(10,float)
    raios=np.zeros(10,float)
    while i<50:
        pts=np.zeros((10,2),int)
        alfa=f.readline()
        nums=alfa.split()
        for j in range(10):
            pts[j,0]=float(nums[j])
        alfa=f.readline()
        nums=alfa.split()
        for j in range(10):
            pts[j,1]=float(nums[j])
        for ii in range(10):
            for jj in range(10):
                raios[jj]=pitag(pts[ii,0],
                    pts[ii,1],pts[jj,0],pts[jj,1])
            maximo[ii]=-999999999999.0
            for jj in range(10):
                if raios[jj]>maximo[ii]:
                    maximo[ii]=raios[jj]
        menor=999999999999.0
        for ii in range(10):
            if maximo[ii]<menor:
                menor=maximo[ii]
                imenor=ii
        ctd=ctd+imenor+1
        i=i+1
    print("So= ",ctd)
```

Para fazer: Há no arquivo acima descrito 50 conjuntos de 10 pares de pontos cada um. Os pontos estão em 2 linhas: na primeira 10 abscissas e na segunda 10 ordenadas, correspondendo o conjunto a 10 pontos. Para cada conjunto, o seu programa deve localizar o ponto central numerando-o de 1 a 10. A resposta a busca é a soma desses 50 números.

\sum nums centro

Histograma Um histograma é uma representação gráfica de algum fenômeno. Feito em colunas, cada uma delas é associada ao valor numérico de alguma medida e a comparação visual imediatamente dá uma boa idéia do comportamento do fenômeno.

Veja-se uma definição: Escreva um programa em Python para ler um vetor de inteiros de tamanho N que representa um histograma normalizado,

com valores variando de 1 a N. O histograma é uma representação gráfica em colunas dos valores do vetor, ou seja, uma distribuição de frequências. Seu programa deve imprimir o histograma a partir do vetor de entrada, conforme exemplos abaixo.

Exemplo de execução (com N=5):

Digite vetor:

1 2 3 4 5

Histograma:

```
@
 @ @
 @ @ @
 @ @ @ @
 @ @ @ @ @
```

Digite vetor:

1 2 5 2 1

Histograma:

```
@
 @
 @
 @ @ @
 @ @ @ @ @
```

```
v=[0]*100
i=ctb=0
while i<50:
    x=f.readline()
    nums=x.split()
    v[0]=int(nums[0])
    for z in range(v[0]):
        v[z+1]=int(nums[z+1])
    for ii in range(v[0]):
        for j in range(v[0]):
            if v[j+1]<(v[0]-ii):
                ctb=ctb+1
        i=i+1
    print("qtd brancos=",ctb)
```

O truque: É imprimir um gráfico a partir de informações numéricas. A solução aqui é imaginar o espaço no papel como uma matriz, que pode ser preenchida com espaço ou com algum caracter, a depender de alguma condição.

Para você No arquivo acima, há 50 vetores de diversos tamanhos (o tamanho do vetor é primeiro número e ele deve ser desconsiderado no processamento do vetor). e que devem gerar histogramas. Totalize em todos eles quantas vezes o comando de impressão de um espaço em branco foi acionado.

\sum espaços



Faça um programa que contabilize os pontos de cada jogador. Seu programa deve ler uma matriz 8x8 representando uma configuração do tabuleiro de xadrez onde cada elemento é um número inteiro que pode ser 0, indicando que não há peça naquela posição, ou um número de 2 dígitos onde o primeiro dígito indica a cor da peça que lá está (1 se for branca e 2 se for preta) e o segundo dígito indica o valor da peça. Por exemplo, 13 indica que naquela posição há um cavalo/bispo branco, já o número 20 indica que a posição tem o rei preto, e assim por diante.

Seu programa deve calcular a pontuação do jogador de brancas e do jogador de pretas e mostrar essa pontuação. Lembre-se que a casa inferior direita de cada jogador é branca.

```
25 0 0 0 0 25 20 0
21 0 29 23 23 21 21 21
0 21 21 23 21 0 0 0
0 0 0 0 0 23 0 0
0 0 19 11 0 0 0 0
0 0 13 0 11 13 13 0
11 11 0 0 13 11 11 11
0 0 15 0 10 0 0 15
Pontuacao Brancas: 22
Pontuacao Pretas: 22
```

```
t=np.zeros((8,8),int)
i=spb=spn=0
while i<100:
    for jj in range(8):
        x=f.readline()
        nums=x.split()
        for kk in range(8):
            t[jj,kk]=int(nums[kk])
        pb=pn=0
        for ii in range(8):
            for jj in range(8):
                cor=t[ii,jj]/10
                pec=t[ii,jj]%10
                if ((ii+jj)%2)==0 and cor==1:
                    pb=pb+pec
                if ((ii+jj)%2)==1 and cor==2:
                    pn=pn+pec
            spb=spb+pb
            spn=spn+pn
            i=i+1
    print("SPB=",spb," SPN=",spn)
```

Para você No arquivo acima, há 100 tabuleiros de 8 linhas por 8 colunas. Devem ser totalizados os pontos de brancas e de negras nos 100 casos.

\sum brancas \sum negras

Para você fazer

\sum nums centro \sum espaços

\sum brancas \sum negras



==== 04/12/2019 10:52:38.6 =====E=PL188p

1 262 2054 1715 1526