

Exercice 1 *Traversée de Matrice*

A) stratégie gloutonne.

```
1 import numpy as np
2
3 M=np.matrix([[1,5,2,5,7,9],[7,3,4,1,2,4],[1,0,4,7,2,1],[2,6,2,1,0,5],[0,1,3,8,9,3],[5,0,7,1,2,5]])
4 #print(M)
5
6 def CheminEtCout(M):
7     n=M.shape[0]
8     i=0
9     j=0
10    chemin=[[i,j]]
11    cout=M[i,j]
12    while ((i,j)!=(n-1,n-1)):
13        #print(i,j)
14        if i==n-1:
15            if j<n-1:
16                j=j+1
17            elif j==n-1:
18                if i<n-1:
19                    i=i+1
20        else:
21            if M[i+1,j]<M[i,j+1]:
22                i=i+1
23            else:
24                j=j+1
25        chemin.append((i,j))
26        cout+=M[i,j]
27    return (chemin,cout)
28
29 print(CheminEtCout(M))
```

B) Stratégie dynamique.

```
1
2 def construire(M):
3     n=M.shape[0]
4     MChoix=np.zeros((n,n))
5     MCout=np.zeros((n,n))
6     MCout[n-1,n-1]=M[n-1,n-1]
7     #dernière ligne et colonne: pas de choix quand
8     #on y est pour les mouvmts ultérieurs
9     for k in range(n-2,-1,-1):
10        MChoix[k,n-1]=-1#on descend
11        MCout[k,n-1]=MCout[k+1,n-1]+M[k,n-1]
12        MChoix[n-1,k]=1#on va à droite
13        MCout[n-1,k]=MCout[n-1,k+1]+M[n-1,k]
14    for i in range(n-2,-1,-1):
15        for j in range(n-2,-1,-1):
16            if MCout[i,j+1]>MCout[i+1,j]:
17                MChoix[i,j]=-1
18                MCout[i,j]=M[i,j]+MCout[i+1,j]
19            else:
20                MChoix[i,j]=1
21                MCout[i,j]=M[i,j]+MCout[i,j+1]
22    return (MCout, MChoix)
23
24
25 print(' Construire (M)=(Mchoix, MCout) '+'\n', construire(M))
26
27
28
29
30 def chemin(M):
31     i=0
32     j=0
33     n=M.shape[0]
34     L=[]
35     (MCout, MChoix)=construire(M)
36     while i+j<2*n-2:
37         if MChoix[i,j]==-1:
38             i=i+1
39             L.append('B')
40         else:
41             j=j+1
42             L.append('D')
43     return(L, MCout[0,0])
44
45
46 print(' chemin (M)=', chemin(M))
```