

DS d'informatique N°1 – Gains en bourse - corrigé

Question 1.

```
def minimum(a):  
    mini = a[0]  
    for v in a:  
        if v < mini:  
            mini = v  
    return mini
```

```
def maximum(a):  
    maxi = a[0]  
    for v in a :  
        if v > maxi:  
            maxi = v  
    return maxi
```

Question 2.

```
def moyenne(a):  
    S=0  
    for v in a:  
        S += v  
    return S/len(a)
```

Question 3.

```
def amplitude(a):  
    return maximum(a) - minimum(a)
```

La fonction maximum contient une boucle qui effectue n itérations, sa complexité est donc $O(n)$. Il en est de même pour la fonction minimum. La complexité de la fonction moyenne est donc $O(n)$ également.

Question 4. Pour le cours $[4,3,2,1]$ le gain maximum est 0 alors que l'amplitude est de 3. L'amplitude représente le gain maximum possible ou la perte maximum possible.

Question 5.

```
def gain(a):  
    maxi = a[1] - a[0]  
    for i in range(len(a)):  
        for j in range(i, len(a)):  
            if a[j] - a[i] > maxi:  
                maxi = a[j] - a[i]  
    return maxi
```

Question 6.

```
def gainModif(a):  
    maxi = 0  
    imin = 0  
    imax = 0  
    delai_min = 0  
    for i in range(len(a)):  
        for j in range(i, len(a)):  
            if (a[j] - a[i] > maxi) or (a[j] - a[i] == maxi and j - i < delai_min):  
                imin = i  
                imax = j
```

```

        delai_min = j - i
        maxi = a[j] - a[i]
    return imin, imax, maxi

```

Question 7.

```

def gain1(a):
    gainMax = 0
    gainCourantMax = 0
    vMin = a[0]
    for i in range(0, len(a)):
        if a[i] < vMin:
            vMin = a[i]
        gainCourantMax = a[i] - vMin
        if gainCourantMax > gainMax:
            gainMax = gainCourantMax
    return gainMax

```

Question 8.

```

def gain1Modif(a):
    gainMax = 0
    gainCourantMax = 0
    iMin = 0           # Indice de la valeur min
    t1 = 0             # Indice de la date d'achat
    duree = 0          # durée entre achat et vente
    for i in range(0, len(a)):
        if a[i] <= a[iMin]:
            iMin = i
        gainCourantMax = a[i] - a[iMin]
        if (gainCourantMax > gainMax) or (gainCourantMax == gainMax and i - iMin < duree):
            gainMax = gainCourantMax
            t1 = iMin
            duree = i - iMin
    return t1, t1 + duree, gainMax

```

Question 9.

```

def gain2(a):
    gainMax = 0
    for i in range(len(a)):
        gain = gain1(a[:i + 1]) + gain1(a[i:])
        if gain > gainMax:
            gainMax = gain
    return gainMax

```

Question 10.

```

def gain2Modif(a):
    gainMax = 0
    iMax = 0           # indice de la séparation optimale
    for i in range(len(a)):
        gain = gain1(a[:i + 1]) + gain1(a[i:])
        if gain > gainMax:
            gainMax = gain
            iMax = i
        i, j = gain1Modif(a[:iMax + 1])[:2]
        ip, jp = gain1Modif(a[iMax:])[2:]
    return i, j, iMax + ip, iMax + jp, gainMax

```