

Concours Polytechnique filières PSI, PT : corrigé

Jean-Loup Carré

Informatique commune – 2002

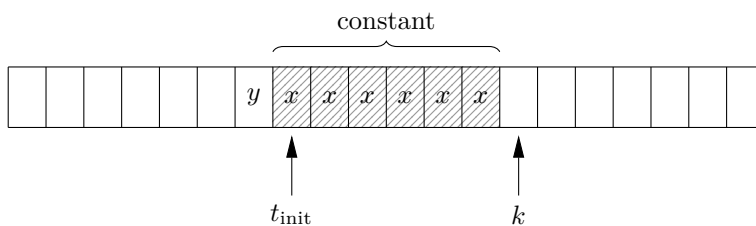
Question 1. **def** compte(x, a):
 c = 0
 for y **in** a:
 if y == x:
 c += 1
 return c

Question 2. **def** occurrences(a):
 r = []
 for k **in** range(len(a)):
 r.append(compte(a[k], a))
 return r

Question 3. Pour calculer la durée maximale, nous avons besoin d'un argument supplémentaire, le tableau t des instants de mesure.

```
def maxconstant(a, t):  
    T_max = 0  
    t_init = t[0]  
    for k in range(1, len(a)):  
        if a[k] == a[k-1]:  
            T_max = max(T_max, t[k] - t_init)  
        else :  
            t_init = t[k]  
    return T_max
```

L'algorithme est correct car la boucle maintient l'invariant suivant : « t_{init} est le minimum des i tels que $t[i : k]$ est un tableau constant », comme l'illustre le dessin ci-dessous (avec $y \neq x$).

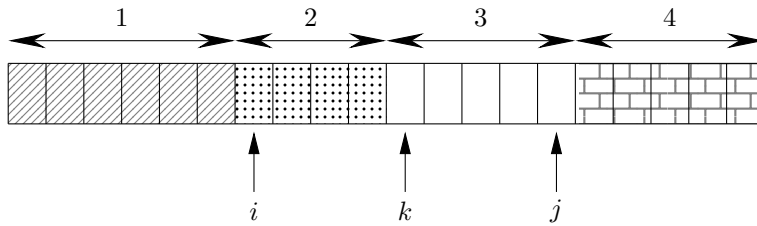


Question 4. **def** maxoccurrences(a, occ):
 i1 = 0
 i2 = -1
 for k **in** range(len(a)):
 if occ[k] > occ[i1]:
 i1, i2 = k, i1
 elif (i2 == -1 **or** occ[i2] < occ[k]) **and** a[k] != a[i1]:
 i2 = k
 print(i1, i2)

Question 5. On introduit une fonction auxiliaire `echange` qui intervertit en même temps deux positions dans les tableaux a et t.

```
def echange(a, t, i, j):  
    a[i], a[j] = a[j], a[i]  
    t[i], t[j] = t[j], t[i]
```

On introduit à présent des indices pour séparer les zones. L'indice i est sur la première case de la zone 1, le k sur la première de la zone 3 et le j sur la dernière case de la zone 3.

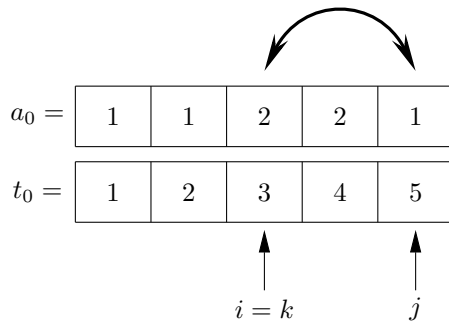


La fonction `trier` va lire le premier élément de la zone 3 (en position k) et va le placer dans la bonne zone (s'il vaut m_1 , on le place dans la zone 1 qu'on étend par exemple).

```
def trier(a, t, m1, m2):
    i = 0
    k = 0
    j = len(a)-1
    while k <= j:
        if a[k] == m1:
            echange(a, t, i, k)
            k += 1
            i += 1
        elif a[k] == m2:
            k += 1
        else:
            echange(a, t, k, j)
            j -= 1
```

Question 6. Appliquons notre fonction `trier` à $a_0 = [1, 1, 2, 2, 1]$ et $t_0 = [1, 2, 3, 4, 5]$.

Lors des 2 premiers passages dans le `while`, nous sommes dans le premier cas (`if`). Nous arrivons dans la situation suivante :



Là nous tombons dans le 3ème cas, et on échange les mesures prises aux temps 3 et 5. Puis l'algorithme se termine sans effectuer aucun échange supplémentaire.

Ainsi, à la fin, a_0 vaut $[1, 1, 1, 2, 2]$ et t_0 vaut $[1, 2, 5, 4, 3]$.

Sur l'exemple, la croissance des dates n'a pas été conservée dans la zone 2. La réponse à la question est donc : NON.