

DM 2 Informatique PSI

8 janvier 2018

On a demandé à des ingénieurs d'écrire une fonction `fusion` qui prend en argument deux tableaux `t1` et `t2` supposés triés dans l'ordre croissant, et qui renvoie un nouveau tableau `t` qui est trié dans l'ordre croissant et qui contient tous les éléments de `t1` et de `t2`. Voici ci-dessous les fonctions qu'ils ont écrites.

- Sans utiliser d'ordinateur, pour chacune des fonctions ci-dessous, dire si elle est correcte ou non.
- Sans utiliser d'ordinateur, pour chaque fonction qui n'est pas correcte, donner un exemple d'appel à la fonction qui montre qu'elle n'est pas correcte et indiquer le résultat obtenu.
- On a aussi demandé que la fonction ait une complexité linéaire en la somme des tailles de `t1` et `t2`. Quelles sont les fonctions qui respectent cette consigne? Plus précisément, donner la complexité de chacune des fonctions données.
- (Facultatif) Vérifier vos réponses à l'aide de l'ordinateur; à faire seulement après avoir fait très sérieusement les questions précédentes!

```
def fusion1(t1,t2):
    t=[]
    i,j=0,0
    while i < len(t1) and j < len(t2):
        if t1[i]<=t2[j]:
            t.append(t1[i])
            i=i+1
        else:
            t.append(t2[j])
            j=j+1
    while i < len(t1):
        t.append(t1[i])
        i=i+1
    while j < len(t2):
        t.append(t2[j])
        j=j+1
    return t
```

```
def fusion3(t1,t2):
    t=[]
    i,j=0,0
    while i < len(t1) and j < len(t2):
        if t1[i]<=t2[j]:
            t.append(t1[i])
            i=i+1
        else:
            t.append(t2[j])
            j=j+1
    if i < len(t1):
        t = t + t1[i:len(t1)]
    else:
        t = t + t2[j:len(t2)]
    return t
```

```
def fusion5(t1,t2):
    if t1==[]:
        return t2
    if t2==[]:
        return t1
    if t1[0] > t2[0]:
        return [t2[0]] + fusion5(t1,t2[1:])
    else:
        return [t1[0]] + fusion5(t1[1:],t2)
```

```
def fusion2(t1,t2):
    t=[]
    i,j=0,0
    while i < len(t1) and j < len(t2):
        if t1[i]<=t2[j]:
            t.append(t1[i])
            i=i+1
        else:
            t.append(t2[j])
            j=j+1
    if i < len(t1):
        t.append(t1[i:len(t1)])
    else:
        t.append(t2[j:len(t2)])
    return t
```

```
def fusion4(t1,t2):
    l1,l2 = len(t1),len(t2)
    i,j = 0,0
    t = []
    for k in range(0,l1+l2):
        if (i<l1 and (j==l2 or t1[i]<=t2[j])):
            t.append(t1[i])
            i=i+1
        else:
            t.append(t2[j])
            j=j+1
    return t
```

```
def fusion6(t1,t2):
    t = t1 + t2
    for j in range(len(t)):
        for i in range (len(t)):
            if t[j] < t[i]:
                t[i],t[j] = t[j],t[i]
    return t
```

```

def fusion7(t1,t2):
    t=[]
    i,j=0,0
    while i < len(t1) and j < len(t2):
        if t1[i]<=t2[j]:
            t.append(t1[i])
            i=i+1
        else:
            t.append(t2[j])
            j=j+1
    if len(t1) > len(t2) :
        for k in range(i,len(t1)):
            t.append(t1[k])
    if len(t1) < len(t2) :
        for k in range(j,len(t2)):
            t.append(t2[k])
    return t

def fusion8(t1,t2):
    t=[]
    for i in range(len(t1)):
        for k in range(len(t2)):
            if t2[k]<=t1[i]:
                t.append(t2[k])
        t.append(t1[i])
    return t

def fusion9(t1,t2):
    t=(len(t1) + len(t2))*[0]
    i,j=0,0
    for k in range(len(t1) + len(t2)):
        if t1[i]<=t2[j]:
            t[k]=t1[i]
            i=i+1
        else:
            t[k]=t2[j]
            j=j+1
    return t

def fusion10(t1,t2):
    t = []
    k = 0
    for i in range(len(t1)):
        for j in range (k,len(t2)):
            if t1[i] >= t2[j]:
                t.append(t2[j])
                k = k+1
        t.append(t1[i])
    return t

def fusion11(t1,t2):
    t = []
    k = 0
    for i in range(len(t1)):
        for j in range (k,len(t2)):
            if t1[i] >= t2[j]:
                t.append(t2[j])
                k = k+1
        t.append(t1[i])
    for j in range (k,len(t2)):
        t.append(t2[j])
    return t

def fusion12(t1,t2):
    t = []
    k = 0
    for i in range(len(t1)):
        while k<len(t2) and t1[i] >= t2[k]:
            t.append(t2[k])
            k = k+1
        t.append(t1[i])
    for j in range (k,len(t2)):
        t.append(t2[j])
    return t

```