TP8: tris

Le document de référence doit être apporté en TP.

Créer un dossier "TP8" dans votre répertoire personnel. Pour l'exercice 1, dans Pyzo, enregistrer un fichier nommé "exo1.py" dans le dossier "TP8". Pour l'exercice 2, dans Pyzo, enregistrer un fichier nommé "exo2.py" dans le dossier "TP8"..... Pour chaque exercice, le fichier python sera régulièrement sauvegardé et éxécuté intégralement avec la commande "Run file as script" du menu Run (Ctrl+Shift+E). Les instructions d'affichage seront saisies sur le fichier python. Au fur et à mesure de l'avancée de l'exercice, les instructions devenant inutiles et gênantes pourront être désactivées en utilisant le caractère #.

Dans ce TP, le terme tableau désigne une liste.

Exercice 1 (Tri d'un tableau par insertion)**.**

Soit t un tableau et h et i deux indices tels que $h \leq i$.
Permuter circulairement $t[h]$,...,$t[i]$ consiste à :

$$t = [\ \ldots\ ,\ v_{h-1}\ ,\ v_h\ ,\ v_{h+1}\ ,\ \ldots\ ,\ v_{i-1}\ ,\ v_i\ ,\ v_{i+1}\ ,\ \ldots\ ]$$

extraire v_i : $t = [\ \ldots\ ,\ v_{h-1}\ ,\ v_h\ ,\ v_{h+1}\ ,\ \ldots\ ,\ v_{i-1}\ ,\ trou\ ,\ v_{i+1}\ ,\ \ldots\ ]$

décaler v_h,...,v_{i-1} : $t = [\ \ldots\ ,\ v_{h-1}\ ,\ trou\ ,\ v_h\ ,\ v_{h+1}\ ,\ \ldots\ ,\ v_{i-1}\ ,\ v_{i+1}\ ,\ \ldots\ ]$

insérer v_i : $t = [\ \ldots\ ,\ v_{h-1}\ ,\ v_i\ ,\ v_h\ ,\ v_{h+1}\ ,\ \ldots\ ,\ v_{i-1}\ ,\ v_{i+1}\ ,\ \ldots\ ]$

Permuter circulairement $t[h]$,...,$t[i]$ revient à effectuer la réaffectation :
$(t[h], t[h+1], ..., t[i-2], t[i-1], t[i]) = (t[i], t[h], ..., t[i-3], t[i-2], t[i-1])$
ce qui, en utilisant une variable auxiliaire x, revient à effectuer dans l'ordre les réaffectations :
$x = t[i]$, $t[i] = t[i-1]$,$t[i-1] = t[i-2]$,$t[i-2] = t[i-3]$,... ,$t[h+1] = t[h]$,$t[h] = x$

Question 1. *Ecrire une fonction nommée "permutation" ayant pour arguments t (tableau) et i et h (deux indices tels que $h \leq i$) et permutant ciculairement $t[h], ..., t[i]$ (effet de bord). On complètera :*

```
def permutation(t,h,i) :
    x=t[i]
    j=i
    while ... :
        t[j]=t[j-1]
        ....
    t[j]=x
```

Attecter à t le tableau t=[3,9,6,7,9,4,5,8,2]. Faire afficher t. Faire éxécuter permutation$(t, 2, 6)$. Faire afficher t.

Présentation de l'algorithme d'insertion :
Donnée : t un tableau de nombres et i un indice.
Précondition : Les termes d'indices au plus $i-1$ sont triés.

Corps :
On note x la valeur de $t[i]$.
| sépare les termes d'indices au plus i et les autres.

$$t = [\underbrace{..., ..., ..., ..., ..., ..., ...}_{termes\text{"}\leq x\text{"}\ (si\ il\ y\ en\ a)}, \underbrace{..., ..., ..., ..., ..., ..., ..., ...}_{termes\text{"}> x\text{"}\ (si\ il\ y\ en\ a)}, x | ..., ..., ..., ...]$$

On permute circulairement les termes suivants :
les termes " $> x$" d'indices au plus $i-1$ et le terme d'indice i

$$t = [\underbrace{...,...,...,...,...,...,...}_{termes\text{"}\leq x\text{"}\ (si\ il\ y\ en\ a)}, x, \underbrace{...,...,...,...,...,...,...,...}_{termes\text{"}> x\text{"}\ (si\ il\ y\ en\ a)} | ...,...,...,...]$$

Effet de bord : t est modifié par permutations circulaires de termes d'indices au plus i.
Postcondition : Les termes d'indices au plus i sont triés.

Question 2. *Ecrire une fonction nommée "insertion" ayant pour arguments t (un tableau de nombres) et i (un indice tel que les termes d'indices au plus $i-1$ sont triés) et réalisant l'algorithme d'insertion (effet de bord). On complètera :*

```
def insertion (t,i) :
    x=t[i]
    j=i
    while ... :
        t[j]=t[j-1]
        ....
    t[j]=x
```

Affecter à t le tableau $[0,2,6,7,9,3,0,3,5]$. Faire afficher t. Faire éxécuter insertion$(t,5)$. Faire afficher t.

Affecter à t le tableau $[6,9,0,2,7,3,0,3,5]$. Faire afficher t. Faire éxécuter insertion$(t,2)$. Faire afficher t.

Présentation de l'algorithme de tri par insertion :
Donnée en entrée : t un tableau de nombres.

Corps :
Pour i variant de 1 à $n-1$:
On exécute insertion(t,i)

Effet de bord : t est modifié par permutations circulaires
Postcondition : t est trié

Exemple d'éxécution de l'algorithme de tri par insertion :
Donnée en entrée : $[9,6,0,2,7,3,0,3,5]$
| *sépare les termes d'indices au plus i des autres*

Début itération $i=1$:	$[9,6\|0,2,7,3,0,3,5]$
Fin itération $i=1$:	$[6,9\|0,2,7,3,0,3,5]$
Début itération $i=2$:	$[6,9,0\|2,7,3,0,3,5]$
Fin itération $i=2$:	$[0,6,9\|2,7,3,0,3,5]$
Début itération $i=3$:	$[0,6,9,2\|7,3,0,3,5]$
Fin itération $i=3$:	$[0,2,6,9\|7,3,0,3,5]$
Début itération $i=4$:	$[0,2,6,9,7\|3,0,3,5]$
Fin itération $i=4$:	$[0,2,6,7,9\|3,0,3,5]$
Début itération $i=5$:	$[0,2,6,9,7,3\|0,3,5]$
Fin itération $i=5$:	$[0,2,3,6,7,9\|0,3,5]$
Début itération $i=6$:	$[0,2,6,9,7,3,0\|3,5]$
Fin itération $i=6$:	$[0,0,2,3,6,7,9\|3,5]$
Début itération $i=7$:	$[0,0,2,3,6,7,9,3\|5]$
Fin itération $i=7$:	$[0,0,2,3,3,6,7,9\|5]$
Début itération $i=8$:	$[0,0,2,3,3,6,7,9,5\|]$
Fin itération $i=8$:	$[0,0,2,3,3,5,6,7,9\|]$

Question 3. *Ecrire une fonction nommée "triinsertion" ayant pour argument t (un tableau de nombres) et triant t par insertion (effet de bord). La fonction triinsertion ne devra pas faire appel à la fonction insertion. A lieu de cela, on copiera le corps de la fonction insertion dans la*

fonction triinsertion. Affecter à la variable t la liste $[9,6,0,2,7,3,0,3,5]$. *Faire afficher t. Exécuter triinsertion(t). Faire afficher t.*

Présentation de l'algorithme de tri par sélection

Donnée : une collection finie d'objets, chaque objet ayant une valeur numérique.

Corps :
On crée une liste vide (liste réceptrice).
Tant que la collection est non vide :
On sélectionne un objet de la collection de valeur minimum.
On extrait cet objet de la collection
et on l'ajoute à la fin de la liste réceptrice.

Donnée en retour : la liste réceptrice
Postcondition : Les termes de la liste réceptrice sont les objets triés par ordre croissant de valeur (car, en notant r la liste réceptrice, n sa longueur et i un indice, $r[i]$ est de valeur minimum parmi $r[i]$,$r[i+1]$,...,$r[n-1]$ donc pour tout indice j supérieur à i, $r[j]$ a valeur supérieure à $r[i]$).

Exemple d'éxécution de l'algorithme de tri par sélection

Donnée : collection (A,15),(B,12),(C,9),(D,20),(E,7),(F,12)
Les objet de la collection sont des couples dont la valeur est la seconde composante.

Liste receptrice :	*Collection* :
[]	$(A,15),(B,12),(C,9),(D,20),(E,7),(F,12)$
$[(E,7)]$	$(A,15),(B,12),(C,9),(D,20),(F,12)$
$[(E,7),(C,9)]$	$(A,15),(B,12),(D,20),(F,12)$
$[(E,7),(C,9),(B,12)]$	$(A,15),(D,20),(F,12)$
$[(E,7),(C,9),(B,12),(F,12)]$	$(A,15),(D,20)$
$[(E,7),(C,9),(B,12),(F,12),(A,15)]$	$(D,20)$
$[(E,7),(C,9),(B,12),(F,12),(A,15),(D,20)]$	*vide*

Exercice 2 (Tri d'un tableau par sélection/échange)**.**
Présentation
Donnée d'entrée : t tableau de nombres

Corps :
On affecte à n la longueur de t.
 # On effectue un tri par sélection de la collection $t[0],...,t[n-1]$ *avec une variable i*
 # telle qu'à chaque étape la liste réceptrice est formée des termes de t d'indices au plus $i-1$
 # et la collection est formée des termes de t d'indices au moins i.
On initialise i à 0.
Tant que $i < n$ *:*
$t = [\underbrace{...,...,...,...,...,...,...,...,...}_{liste\,receptrice\,(vide\,si\,i=0)}, \underbrace{t[i],t[i+1],...,t[n-1]}_{collection}]$
On sélectionne $t[k]$ *de valeur minimum dans la collection* $t[i],t[i+1],...,t[n-1]$
On échange $t[i]$ *et* $t[k]$.
$t = [\underbrace{...,...,...,...,...,...,...,...,...}_{liste\,receptrice\,(vide\,si\,i=0)}, \underbrace{t[i],t[i+1],...,t[n-1]}_{collection}]$
$t[i]$ *est de valeur minimum dans la collection*
 # On extrait $t[i]$ *de la collection et on l'ajoute à la fin de la liste réceptrice ainsi :*
On réaffecte à i la valeur de $i+1$.

Effet de bord : t est modifié par une suite d'échanges
Postcondition : t est trié

(car, à la fin, $i = n$ donc la collection est vide et t est égal à la liste réceptrice qui est triée)

Remarque : la dernière itération est inutile car, au début de cette itération, la collection ne contient que $t[n-1]$ donc $k = n-1 = i$ donc l'échange de $t[i]$ et $t[k]$ est inutile. On préfère donc une boucle "Tant que $i < n-1$".

Exemple d'éxécution :

Donnée en entrée : $t = [9, 6, 1, 2, 7]$

| sépare la liste réceptrice et la collection
Initialisation : $i = 0$, $t = [|9, 6, 1, 2, 7]$
Première itération :
Après sélection : $k = 2$.
Après échange : $t = [|1, 6, 9, 2, 7]$.
Après extraction/ajout : $i = 1$, $t = [1|6, 9, 2, 7]$
Seconde itération :
Après sélection : $k = 3$.
Après échange : $t = [1|2, 9, 6, 7]$
Après extraction/ajout : $i = 2$, $t = [1, 2|9, 6, 7]$
Troisième itération :
Après sélection : $k = 3$:
Après échange : $t = [1, 2|6, 9, 7]$
Après extraction/ajout : $i = 3$, $t = [1, 2, 6|9, 7]$
Quatrième itération :
Après sélection : $k = 4$
Après échange : $t = [1, 2, 6|7, 9]$
Après extraction/ajout : $i = 4$, $t = [1, 2, 6, 7|9]$
Dernière itération inutile.

Question 1. *Ecrire une fonction nommée "argminpart" ayant pour arguments t (tableau de nombres) et i (indice du tableau) et retournant un indice k avec $k \geq i$ tel que $t[k]$ est minimum. Faire afficher $argminpart([1, 2, 9, 6, 7], 2)$.*

L'instruction $t[i], t[k] = t[k], t[i]$ permet d'échanger deux termes d'un tableau t.

Question 2. *Ecrire une fonction nommée "triselectionechange" ayant pour argument t (tableau de nombres) et triant t par sélection/échange (effet de bord). Le fonction "triselectionechange" ne devra pas faire appel à la fonction "argminpart". Au lieu de cela, on copiera la majeure partie du corps de la fonction "argminpart" dans le corps de la fonction "triselectionechange". Affecter à la variable t la liste $[9, 6, 0, 2, 7]$. Faire afficher t. Exécuter $triselectionechange(t)$. Faire afficher t.*

Exercice 3 (Tri fusion (ou tri dichotomique))**.**
Algorithme de fusion de deux tableaux triés
Données en entrée : $t1$ et $t2$ tableaux de nombres.
Précondition : $t1$ et $t2$ sont triés

Corps :
On affecte à $n1$ et $n2$ les longueurs de $t1$ et $t2$.
 # On effectue un tri par sélection de la collection $t1[0], ...t1[n1-1], t2[0], ...t2[n2-1]$
 # avec deux variables $i1$ et $i2$ telles que qu'a chaque étape la collection est formée
 # des termes de $t1$ d'indices au moins $i1$ et des termes de $t2$ d'indices au moins $i2$.
On initialise $i1$ et $i2$ à 0.
On intialise r à la liste vide.
 # r liste réceptrice
Tant que $i1 < n1$ et $i1 < n2$:
    # La collection est $t1[i1], ...t1[n1-1], t2[i2], ..., t2[n2-1]$

 # Puisque $t1$ et $t2$ sont triés, $t1[i1]$ ou $t2[i2]$ est un minimum de la collection.
Si $t1[i1] \leq t2[i2]$:
 # $t1[i1]$ est un minimum de la collection
On ajoute $t1[i1]$ à la fin de r.
 # On extrait $t1[i1]$ de la collection ainsi :
On réaffecte à $i1$ la valeur de $i1 + 1$
Sinon :
 # $t2[i2]$ est un minimum de la collection.
On ajoute $t2[i2]$ à la fin de r.
On réaffecte à $i2$ la valeur de $i2 + 1$
Premier cas : $i1 = n1$
 # Tous les termes de $t1$ ont été extraits de la collection
Tant que $i2 < n2$:
 # La collection est égale à $t2[i2], ..., t2[n2 - 1]$
 # Puisque $t2$ est trié, $t2[i2]$ est minimum de la collection
On ajoute $t2[i2]$ à la fin de r.
On réaffecte à $i2$ la valeur de $i2 + 1$
Second cas : $i2 = n2$
Tant que $i1 < n1$:
On ajoute $t1[i1]$ à la fin de r.
On réaffecte à $i1$ la valeur de $i1 + 1$

Donnée en retour : r tableau de nombre.
Postcondition : r est formé des termes de $t1$ et $t2$ triés par ordre croissant.

Question 1. *Ecrire une fonction nommée "fusion" ayant pour arguments $t1$ et $t2$ (deux tableaux de nombres triés) et retournant r tableau de nombres formé des termes de $t1$ et de $t2$ triés par ordre croissant. Faire afficher $fusion([0, 2, 6, 9], [0, 3, 3, 5, 7])$.*

Algorithme récursif de tri fusion (ou tri dichotomique) :
Donnée en entrée : t un tableau de nombres

Corps :
Affecter à n la longueur de t.
Premier cas : $n \geq 2$
 # Principe de la dichotomie : découper t en deux moitiés $t1$ et $t2$
Affecter à m la valeur $n//2$.
Affecter à $t1$ le tableau formé des termes $t[0], ...t[m - 1]$.
Affecter à $t2$ le tableau formé des termes $t[m]$,...,$t[n - 1]$.
Réaffecter à $t1$ le tableau obtenu par tri fusion de $t1$.
Réaffecter à $t2$ le tableau obtenu par tri fusion de $t2$.
Affecter à r la fusion des tableau triés $t1$ et $t2$.
Retourner r.
Second cas : $n \leq 1$
 # t a au plus un terme donc est déjà trié
Retourner t

Donnée en retour : r tableau de nombres
Postcondition : r est formé des termes de t triés

Question 2. *Ecrire une fonction nommée "tableau" ayant pour argument t (un tableau) et a et b (deux indices tels que $a \leq b$) et retournant s le tableau formé des termes $t[a], t[a + 1], ..., t[b]$. Faire afficher $tableau([9, 6, 0, 2, 7, 3, 0, 3, 5], 0, 3)$ et $tableau([9, 6, 0, 2, 7, 3, 0, 3, 5], 4, 8)$.*

Question 3. *Ecrire une fonction récursive nommée "trifusion" ayant pour argument t (un tableau de nombres), réalisant l'algorithme de tri fusion, et retournant r un tableau formé des termes de t triés. Faire afficher $trifusion([9, 6, 0, 2, 7, 3, 0, 3, 5])$.*

Remarque.

La création d'un tableau nécessite d'allouer de la mémoire. L'éxécution de la fonction "trifusion" de l'exercice précédent entraine de nombreux appels de la fonction "tableau" donc nécessite d'allouer beaucoup de mémoire. Nous allons améliorer cela :

Exercice 4 (Tri fusion amélioré)**.**

Soit x et y deux tableaux de même longueur et a, m, b trois indices tels $a \leq m < b$ tels que $x[a], ..., x[m]$ sont triés et $x[m+1], .., x[b]$ sont triés. On appelle fusion de $x[a], ..., x[m]$ et $x[m+1], ..., x[b]$ sur y le tri par sélection de la collection $x[a], ..., x[m], x[m+1], ..., x[b]$ avec la liste de réception formée des termes $y[a]$,...,$y[b]$ en fin de tri. Ceci nécessite évidemment de réaffecter $y[a]$,...,$y[b]$ (effet de bord).

Question 1. *Ecrire une fonction "fusion" ayant pour arguments x et y (deux tableaux de même longueur) et a,m,b (indices tels que $a \leq m < b$ et $x[a], ..., x[m]$ sont triés et $x[m+1], ..., x[b]$ sont triés) effectuant la fusion de $x[a], ..., x[m]$ et $x[m+1], ..., x[b]$ sur y (effet de bord). Affecter à x et y les tableaux $[5, 2, 8, 1, 2, 6, 9, 1, 3, 3, 5, 7, 6, 9]$ et $[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$. Faire afficher x et y. Exécuter $fusion(x, y, 3, 6, 11)$. Faire afficher x et y.*

Soit x et y deux tableaux de même longueur et a et b deux indices tels que $a \leq b$ et $x[a] = y[a]$,...,$x[b] = y[b]$. Le tri fusion de $y[a], ..., y[b]$ depuis x est l'algorithme récursif suivant :

Premier cas : $a \neq b$

On pose $m = (a + b)//2$

On effectue le tri fusion de $x[a], ..., x[m]$ depuis y.

On effectue le tri fusion de $x[m+1], ..., x[b]$ depuis y.

On effectue la fusion de $x[a], ..., x[m]$ et $x[m+1], ..., x[b]$ sur y.

Second cas : $a = b$.

Il n'y a rien à faire.

Question 2. *Ecrire une fonction récursive nommée "trifusion" ayant pour arguments x et y (deux tableaux de même longueur) et a et b (deux indices tels que $a \leq b$ et $x[a] = y[a]$,...,$x[b] = y[b]$) et effectuant le tri fusion de $y[a], ..., y[b]$ depuis x (effet de bord).*

Soit t un variable désignant un tableau. On effectue l'instruction $s = t$. Attention : s n'est pas une copie de t. s et t sont deux noms de variables qui désignent le même tableau.

Question 3. *Recopier et faire éxécuter la suite d'instructions suivante et examiner les affichages afin de s'assurer d'avoir compris le fonctionnement.*

$t = [2, 4, 8, 7, 2, 5]$	$t[2] = 3$
$s = t$	$print(t, s)$

Le tri fusion d'un tableau t est le tri fusion de tous les termes de t depuis s, avec s copie de t.

Question 4. *Coller la fonction "tableau" (après l'avoir copiée dans le fichier exo3.py). Ecrire une fonction nommée "trifusionprinc" ayant pour argument t (un tableau) et effectuant le tri fusion de t (effet de bord). Dans le corps de "trifusionprinc", on appellera la fonction "tableau" pour copier t. Affecter à t le tableau $[9, 6, 0, 2, 7, 3, 0, 3, 5]$. Faire afficher t. Exécuter $trifusionprinc(t)$. Faire afficher t.*

Remarque.

L'éxécution de trifusionprinc(t) ne nécessite qu'un seul appel de la fonction "tableau" pour copier t au début.