

Tri d'une séquence

Questions : Un annuaire est représenté ici par une liste de quadruplets (nom, age, ville, téléphone).

Exemple d'annuaire :

```
item1 = ('dupont', 23, 'brest', '06789656')
item2 = ('abgral', 61, 'lille', '06231298')
item3 = ('dupont', 23, 'brest', '02989656')
item4 = ('abgral', 67, 'brest', '06556438')
item5 = ('martin', 38, 'paris', '01674523')
item6 = ('abgral', 67, 'lille', '06231298')
```

```
annuaire = [item1, item2, item3, item4, item5, item6]
```

Trier l'annuaire, par ordre croissant ou décroissant, selon des critères donnés par une liste des clés successives. L'ordre des critères est précisé par une liste des rangs successifs des champs du quadruplet (nom, age, ville, téléphone). Par exemple, la liste [3,0,2,1] indique qu'il faut d'abord (clé primaire) trier selon les numéros de téléphone (champ n° 3 dans le quadruplet), puis (clé secondaire) selon les noms (champ n° 0 dans le quadruplet), puis selon la ville (champ n° 2 dans le quadruplet) et enfin selon les âges (champ n° 1 dans le quadruplet).

Méthode : On utilise un tri (par exemple le tri par insertion) où la fonction `cmp` de comparaison est passée en argument (ordre croissant par défaut : `cmp = (lambda x,y : x < y)`).

```
1 def triInsertion(t,cmp=(lambda x,y : x < y)) :
2     assert type(t) is list
3     for i in range(1,len(t)):
4         x = t[i]
5         k = i
6         for j in range(i-1,-1,-1):
7             if not cmp(t[j],x) :
8                 k = k-1
9                 t[j+1] = t[j]
10        t[k] = x
11    return
```

Dans le cas de l'annuaire, il faut adapter la fonction de comparaison pour tenir compte des clés successives de comparaison ([0,1,2,3] par défaut).

```
1 def cmpItems(i1,i2,cles=[0,1,2,3],cmp=(lambda x,y : x < y)) :
2     assert type(i1) is tuple and len(i1) == 4
3     assert type(i2) is tuple and len(i2) == 4
4     assert type(cles) is list and (0 < len(cles) <= 4)
5     for k in cles:
6         if cmp(i1[k],i2[k]) : return True
7         elif i1[k] == i2[k] : pass
8         else : return False
9     return False
```

Il reste à adapter l'appel de la fonction de tri à chaque cas particulier :

```
>>> cles = [3, 0, 1, 2]
```

```
>>> decroissant = (lambda x,y : x > y)
>>> triInsertion(annuaire,(lambda x,y : cmpItems(x,y,cles,decroissant)))
```

Réponses :

1. [3, 0, 1, 2], croissant
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '02989656')
('abgral', 61, 'lille', '06231298')
('abgral', 67, 'lille', '06231298')
('abgral', 67, 'brest', '06556438')
('dupont', 23, 'brest', '06789656')
2. [3, 0, 2, 1], croissant
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '02989656')
('abgral', 61, 'lille', '06231298')
('abgral', 67, 'lille', '06231298')
('abgral', 67, 'brest', '06556438')
('dupont', 23, 'brest', '06789656')
3. [3, 1, 0, 2], croissant
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '02989656')
('abgral', 61, 'lille', '06231298')
('abgral', 67, 'lille', '06231298')
('abgral', 67, 'brest', '06556438')
('dupont', 23, 'brest', '06789656')
4. [3, 1, 2, 0], croissant
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '02989656')
('abgral', 61, 'lille', '06231298')
('abgral', 67, 'lille', '06231298')
('abgral', 67, 'brest', '06556438')
('dupont', 23, 'brest', '06789656')
5. [3, 2, 0, 1], croissant
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '02989656')
('abgral', 61, 'lille', '06231298')
('abgral', 67, 'lille', '06231298')
('abgral', 67, 'brest', '06556438')
('dupont', 23, 'brest', '06789656')
6. [3, 2, 1, 0], croissant
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '02989656')
('abgral', 61, 'lille', '06231298')
('abgral', 67, 'lille', '06231298')
('abgral', 67, 'brest', '06556438')
('dupont', 23, 'brest', '06789656')
7. [1, 0, 2, 3], décroissant
('abgral', 67, 'lille', '06231298')
('abgral', 67, 'brest', '06556438')
('abgral', 61, 'lille', '06231298')
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '06789656')
('dupont', 23, 'brest', '02989656')
8. [1, 0, 3, 2], décroissant
('abgral', 67, 'brest', '06556438')
('abgral', 67, 'lille', '06231298')
('abgral', 61, 'lille', '06231298')
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '06789656')
('dupont', 23, 'brest', '02989656')
9. [1, 2, 0, 3], décroissant
('abgral', 67, 'lille', '06231298')
('abgral', 67, 'brest', '06556438')
('abgral', 61, 'lille', '06231298')
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '06789656')
('dupont', 23, 'brest', '02989656')
10. [1, 2, 3, 0], décroissant
('abgral', 67, 'lille', '06231298')
('abgral', 67, 'brest', '06556438')
('abgral', 61, 'lille', '06231298')
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '06789656')
('dupont', 23, 'brest', '02989656')
11. [1, 3, 0, 2], décroissant
('abgral', 67, 'brest', '06556438')
('abgral', 67, 'lille', '06231298')
('abgral', 61, 'lille', '06231298')
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '06789656')
('dupont', 23, 'brest', '02989656')
12. [1, 3, 2, 0], décroissant
('abgral', 67, 'brest', '06556438')
('abgral', 67, 'lille', '06231298')
('abgral', 61, 'lille', '06231298')
('martin', 38, 'paris', '01674523')
('dupont', 23, 'brest', '06789656')
('dupont', 23, 'brest', '02989656')

13. [2, 0, 1, 3], croissant
 ('abgral', 67, 'brest', '06556438')
 ('dupont', 23, 'brest', '02989656')
 ('dupont', 23, 'brest', '06789656')
 ('abgral', 61, 'lille', '06231298')
 ('abgral', 67, 'lille', '06231298')
 ('martin', 38, 'paris', '01674523')
14. [2, 0, 3, 1], croissant
 ('abgral', 67, 'brest', '06556438')
 ('dupont', 23, 'brest', '02989656')
 ('dupont', 23, 'brest', '06789656')
 ('abgral', 61, 'lille', '06231298')
 ('abgral', 67, 'lille', '06231298')
 ('martin', 38, 'paris', '01674523')
15. [2, 1, 0, 3], croissant
 ('dupont', 23, 'brest', '02989656')
 ('dupont', 23, 'brest', '06789656')
 ('abgral', 67, 'brest', '06556438')
 ('abgral', 61, 'lille', '06231298')
 ('abgral', 67, 'lille', '06231298')
 ('martin', 38, 'paris', '01674523')
16. [2, 1, 3, 0], croissant
 ('dupont', 23, 'brest', '02989656')
 ('dupont', 23, 'brest', '06789656')
 ('abgral', 67, 'brest', '06556438')
 ('abgral', 61, 'lille', '06231298')
 ('abgral', 67, 'lille', '06231298')
 ('martin', 38, 'paris', '01674523')
17. [2, 3, 0, 1], croissant
 ('dupont', 23, 'brest', '02989656')
 ('abgral', 67, 'brest', '06556438')
 ('dupont', 23, 'brest', '06789656')
 ('abgral', 61, 'lille', '06231298')
 ('abgral', 67, 'lille', '06231298')
 ('martin', 38, 'paris', '01674523')
18. [2, 3, 1, 0], croissant
 ('dupont', 23, 'brest', '02989656')
 ('abgral', 67, 'brest', '06556438')
 ('dupont', 23, 'brest', '06789656')
 ('abgral', 61, 'lille', '06231298')
 ('abgral', 67, 'lille', '06231298')
 ('martin', 38, 'paris', '01674523')
19. [0, 1, 2, 3], décroissant
 ('martin', 38, 'paris', '01674523')
 ('dupont', 23, 'brest', '06789656')
 ('dupont', 23, 'brest', '02989656')
 ('abgral', 67, 'lille', '06231298')
 ('abgral', 67, 'brest', '06556438')
 ('abgral', 61, 'lille', '06231298')
20. [0, 1, 3, 2], décroissant
 ('martin', 38, 'paris', '01674523')
 ('dupont', 23, 'brest', '06789656')
 ('dupont', 23, 'brest', '02989656')
 ('abgral', 67, 'brest', '06556438')
 ('abgral', 67, 'lille', '06231298')
 ('abgral', 61, 'lille', '06231298')
21. [0, 2, 1, 3], décroissant
 ('martin', 38, 'paris', '01674523')
 ('dupont', 23, 'brest', '06789656')
 ('dupont', 23, 'brest', '02989656')
 ('abgral', 67, 'lille', '06231298')
 ('abgral', 61, 'lille', '06231298')
 ('abgral', 67, 'brest', '06556438')
22. [0, 2, 3, 1], décroissant
 ('martin', 38, 'paris', '01674523')
 ('dupont', 23, 'brest', '06789656')
 ('dupont', 23, 'brest', '02989656')
 ('abgral', 67, 'lille', '06231298')
 ('abgral', 61, 'lille', '06231298')
 ('abgral', 67, 'brest', '06556438')
23. [0, 3, 1, 2], décroissant
 ('martin', 38, 'paris', '01674523')
 ('dupont', 23, 'brest', '06789656')
 ('dupont', 23, 'brest', '02989656')
 ('abgral', 67, 'brest', '06556438')
 ('abgral', 67, 'lille', '06231298')
 ('abgral', 61, 'lille', '06231298')
24. [0, 3, 2, 1], décroissant
 ('martin', 38, 'paris', '01674523')
 ('dupont', 23, 'brest', '06789656')
 ('dupont', 23, 'brest', '02989656')
 ('abgral', 67, 'brest', '06556438')
 ('abgral', 67, 'lille', '06231298')
 ('abgral', 61, 'lille', '06231298')