

Activité découverte de la récursivité

Année de terminale

Objectifs

Réinvestir le problème du *crêpier psychorigide*, étudié lors de la classe de première, pour introduire la notion de *fonction récursive* de la classe de terminale.

1 Rappel du problème

1.1 Algorithme

Un crêpier souhaite ordonner une pile de crêpe en fonction de leur taille, en ne pouvant exécuter comme seule opération que glisser sa spatule sous une crêpe et retourner la pile au dessus.

Pour résoudre ce problème nous avons déterminé l'algorithme suivant :

1. trouver la position de la plus grande crêpe ;
2. mettre la spatule sous la plus grande crêpe et retourner le haut de la pile ;
3. mettre la spatule tout en bas et retourner l'intégralité des crêpes ;
4. recommencer la procédure en ignorant les crêpes rangées.

1.2 Programmation : rappel

Informatiquement, ce problème est équivalent à trier une liste d'entier.

L'étape **1.** de l'algorithme demande de repérer l'indice du plus grand entier, dans la partie de liste qui n'a pas déjà été triée (sinon il repérera systématiquement la plus grande crêpe, à la fin de la liste après la première itération).

La fonction que nous avons utilisé est :

```
1 def indice_grande_crepe(pile, index):
2     indice_gc = 0
3     for i in range(index+1):
4         if pile[i] > pile[indice_gc]:
5             indice_gc = i
6     return indice_gc
```

Les étapes **2.** et **3.** demandent de retourner le début de la pile. Des méthodes implémentées dans Python permettent de le faire très simplement, mais afin de mieux appréhender la complexité de l'algorithme, nous avons écrit la fonction suivante :

```
1 def rotation_liste(pile, index):
2     nombre_inversions = (index + 1) // 2
3     for i in range(nombre_inversions):
4         j = index - i
5         pile[i], pile[j] = pile[j], pile[i]
6     return pile
```

2 Récursivité

2.1 Présentation

En première, nous avons codé le programme de façon itérative. Mais une façon d'expliquer l'instruction **4.** de l'algorithme est de recommencer les opérations **1.** à **3.** sur la pile des crêpes non encore triées (c'est à dire toutes sauf celle du bas la première fois qu'on recommence l'algorithme, toutes sauf les deux du bas la seconde fois, etc.).

Cette façon de décrire l'algorithme, en le rappelant plusieurs fois sur un ensemble à chaque fois plus petit, est une *démarche récursive*.

2.2 Structure générale des fonctions récursives

Les fonctions récursives s'appellent elle-même, mais avec un argument à chaque fois plus simple (ici, une liste dont la longueur diminuera de un à chaque appel).

La terminaison sera assurée par le traitement du cas trivial (ici, le tri d'une liste contenant un seul entier).

2.3 Programmation

Compléter le code suivant, afin de résoudre le problème du crêpier psychorigide à l'aide d'une fonction récursive, puis tester votre programme.

```
1 def tri_crepe_recuratif(pile):
2     if len(pile) == 1:
3         return pile
4     indice_gc = indice_grande_crepe(pile, len(pile)-1)
5     pile = rotation_liste("a_compléter")
6     pile = rotation_liste("a_compléter")
7     return tri_crepe_recuratif("a_compléter") + "a_compléter"
```

Solution : code complet, avec un test.

crepe_psychorigide_recuratif_commentaires.py

```
1 def tri_crepe_recuratif(pile):
2     """
3     Fonction qui reçoit en argument une liste d'entiers non
4     triés, et qui retourne la liste triée par ordre croissant
5     par fonction récursive.
6     """
7     if len(pile) == 1:
8         return pile
9     indice_gc = indice_grande_crepe(pile, len(pile)-1)
10    pile = rotation_liste(pile, indice_gc)
11    pile = rotation_liste(pile, len(pile)-1)
12    return tri_crepe_recuratif(pile[:len(pile) - 1]) + [pile[-1]]
13
14
15 def indice_grande_crepe(pile, index):
16     """
17     Fonction qui reçoit en argument une liste d'entiers non
18     triés et un entier index qui correspond à l'indice du
19     dernier élément de la liste non triée.
20     La fonction retourne l'indice de la plus grande crêpe
21     de la liste.
22     """
23     indice_gc = 0
24     for i in range(index+1):
25         if pile[i] > pile[indice_gc]:
26             indice_gc = i
27     return indice_gc
28
29
30 def rotation_liste(pile, index):
31     """
32     Glisser la spatule sous la crêpe au bon index et retourner
33     le tableau sélectionné pour mettre la grande crêpe en haut
34     """
35     nombre_inversions = (index + 1) // 2
36     for i in range(nombre_inversions):
37         j = index - i
38         pile[i], pile[j] = pile[j], pile[i]
39     return pile
40
41
42
43 if __name__ == '__main__':
44     #les nombres correspondent à la taille des crêpes
45     pile = [1, 4, 5, 2, 12, 8, 6, 7, 9, 3]
46     print("\ncrêpes_non_triées:")
47     print(pile)
48     print("\nLes_itérations:")
49     pile = tri_crepe_recuratif(pile)
50     print("\nCrêpes_triées:", pile)
```