

EXERCICE 1 (6 points)

Cet exercice porte sur les algorithmes de tri et la programmation en Python.

On se propose dans cet exercice de se pencher sur un algorithme pour trier un tableau appelé le tri de *Stooge*.

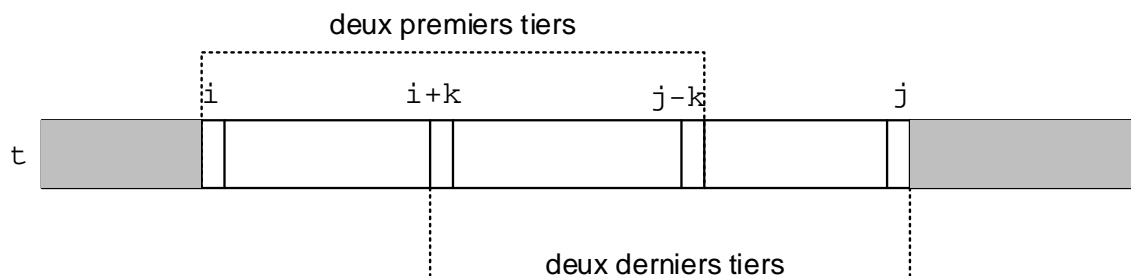
Pour trier les éléments situés entre les indices i et j , où $i < j$, dans un tableau t par ce tri, on procède ainsi :

- si les éléments d'indice i et j sont mal placés, on les échange ;
- si il y a au moins trois éléments entre les indices i et j
 - on trie les deux premiers tiers du tableau avec cette méthode ;
 - on trie les deux derniers tiers du tableau avec cette méthode ;
 - on trie à nouveau les deux premiers tiers du tableau avec cette méthode.

Pour réaliser ce découpage en tiers, on considère l'entier k défini par l'expression

$$(j - i + 1) // 3,$$

et on considère les indices intermédiaires $i+k$ et $j-k$.



Voici le code partiel de l'algorithme du tri de Stooge en Python qui trie donc les éléments d'un tableau par ordre croissant.

```
1 def triStooge(tab, i, j):
2     if tab[i] > tab[j]:
3         echange(tab, i, j)
4     if (j - i) > 1:
5         k = (j - i + 1) // 3
6         triStooge(...)
7         triStooge(...)
8         triStooge(...)
```

1. Écrire la fonction `echange(tab, i, j)` qui prend en arguments une liste Python `tab` et deux indices `i`, `j`, et réalise sur place l'échange des valeurs dans `tab` à ces indices. La fonction ne renvoie rien.

- Finaliser le programme précédent en complétant les lignes 6, 7 et 8 sur votre copie.
- Indiquer en le justifiant si cet algorithme est itératif ou récursif.

Soit l'appel `triStooge(A, 0, 5)` avec $A = [5, 6, 4, 2, 3, 1]$.

- Déterminer la valeur numérique prise par k lors de ce premier appel. Une justification est attendue.

La figure ci-dessous présente l'arbre (incomplet) des appels récursifs effectués depuis l'appel `triStooge(A, 0, 5)`.

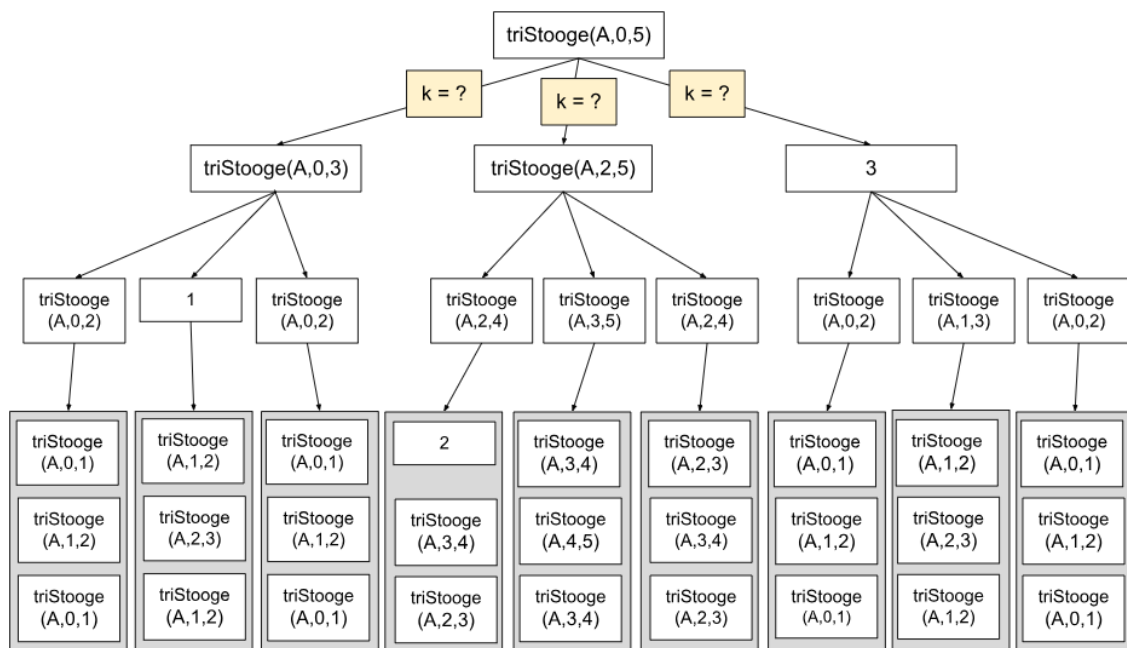


Figure 1. Arbre des appels récursifs pour `triStooge(A, 0, 5)`

- Dénombrer le nombre d'appels récursifs effectués lors du tri sans compter l'appel initial.
- Déterminer les appels effectués dans les cases 1, 2 et 3 de cet arbre des appels récursifs.
- Dans cette question, $A = [5, 4, 6, 2]$. Recopier le tableau ci-dessous et le compléter (remplacer les ??).

Appel	Valeur de A avant l'appel	Valeur de A après l'appel
<code>triStooge(A, 0, 3)</code>	$[5, 6, 4, 2]$??
??	$[2, 6, 4, 5]$??
<code>triStooge(A, 1, 3)</code>	$[2, 4, 6, 5]$??

Appel	Valeur de A avant l'appel	Valeur de A après l'appel
??	[2, 4, 5, 6]	[2, 4, 5, 6]

On montre que le coût en temps dans le pire des cas de l'algorithme de Stooage est de l'ordre de n^e , avec e environ égal à $\frac{8}{3}$.

8. Donner un algorithme de tri dont le coût est strictement meilleur.