

# Devoir Surveillé 2

15 Janvier 2021 8h30-10h30

## Algorithmes de Tris

1. Expliquez en quoi consiste un algorithme de tri.
2. On teste trois fonctions `fonction_a`, `fonction_b`, `fonction_c` sur différents exemples ci-dessous. Expliquez pour chacune d'entre-elles si elles peuvent être des fonctions de tri ou pas.

```
1  L_1 = [14, 1, 4, 3]
2  L_2 = [1, 30, 3.141592, 10**23]
3  L_3 = [1, 1, 2, 3, 5]
4
5
6  fonction_a(L_1)
7  => [14, 4, 3, 1]
8
9  fonction_a(L_2)
10 => [10**23, 30, 3.141592, 1]
11
12 fonction_a(L_3)
13 => [5, 3, 2, 1, 1]
14
15 fonction_b(L_1)
16 => [1, 3, 4, 14]
17
18 fonction_b(L_2)
19 => [1, 30, 3.141592, 10**23]
20
21 fonction_b(L_3)
22 => [1, 1, 2, 3, 5]
23
24 fonction_c(L_1)
25 => [1, 3, 4, 14]
26
27 fonction_c(L_2)
28 => [30, 3.141592, 10**23, 1]
29
30 fonction_c(L_3)
31 => [1, 1, 2, 3, 5]
32
```

3. On donne l'algorithme suivant :

```
1  def fonction_1(L,debut,fin):
2      p = L[debut]
3      j = debut
4      for i in range(debut,fin):
```

```

5         if p > L[i] :
6             L = L[:j]+[L[i]]+L[j:i]+L[i+1:]
7             j +=1
8         return j,L
9
10    def fonction_2(L,debut,fin):
11        if debut+1>=fin :
12            return L
13        else :
14            j,L = fonction_1(L,debut,fin)
15            debut_1 = debut
16            fin_1 = j
17            L = fonction_2(L,debut_1,fin_1)
18            debut_2 = j+1
19            fin_2 = fin
20            L = fonction_2(L,debut_2,fin_2)
21        return L
22
23    def fonction_3(L) :
24        L = fonction_2(L,0,len(L))
25        return L
26

```

De quel algorithme s'agit-il? Justifier votre réponse en expliquant des caractéristiques propres au fonctionnement de cet algorithme. Dans la suite de l'énoncé on nommera cet algorithme de tri, le tri A.

4. Un autre algorithme de tri que le tri A, utilise une technique dites de "diviser pour régner", expliquer la signification de cette expression et nommer l'autre algorithme de tri. Dans la suite de l'énoncé on nommera cet algorithme de tri, le tri B.
5. Expliquez le principe d'un troisième algorithme de tri (que l'on nommera tri C). On pourra s'aider de l'exemple des différentes étapes du fonctionnement de tri C sur la liste [14, 1, 4, 3].
6. Écrire en python la fonction qui réalise le tri B.
7. Pour chaque algorithmes de tri A, B, et C, sont-ils récursifs ou itératifs?

## Performances des algorithmes

8. Pour comparer la performance des algorithmes de tri, on utilise deux critères. Donner ces critères et à quoi correspondent ils?
9. Lorsqu'on évalue la performance d'un algorithme de tri, on parle parfois de meilleurs des cas et de pire des cas. Définir le terme de meilleur et pire des cas.
10. Pour chaque algorithme de tri A, B, et C, dire si un critère de performance admet un meilleur des cas ou un pire des cas, si oui, expliquer quel est le meilleur des cas et le pire des cas.
11. Pour les graphes ci-dessous on a oublié de noter le nom du tri. Associer un tri A, B, ou C à chaque graphe, en justifiant votre réponse.

figure 1

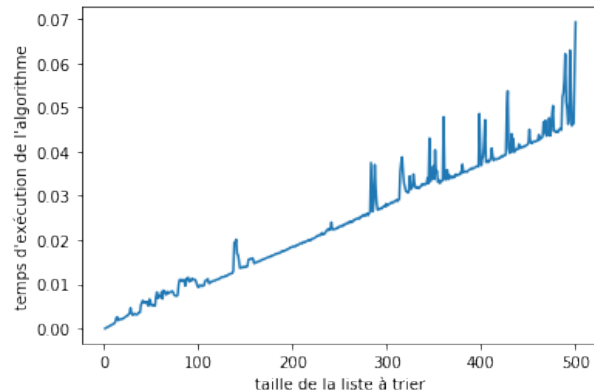


figure 2

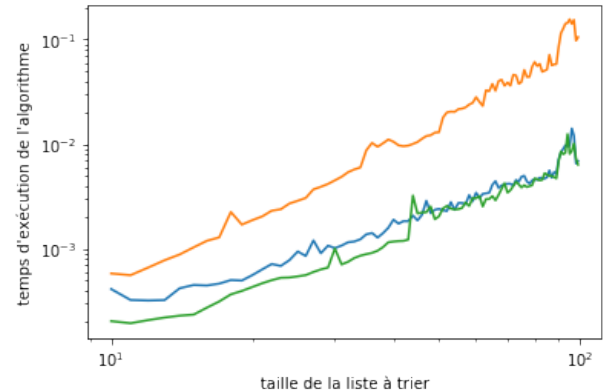
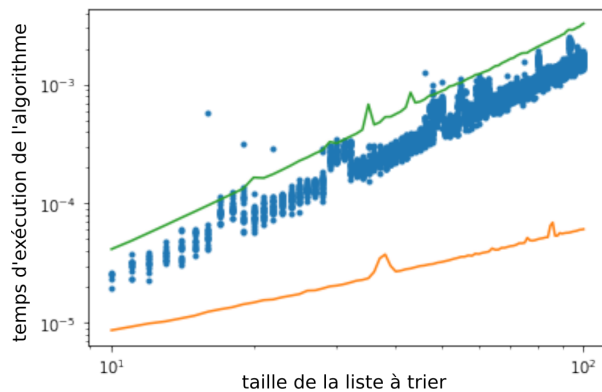


figure 3



12. Certaines versions d'écriture d'algorithme sont dites "en place" et d'autres non. Définir le terme de tri en place.
13. Quel critère de performance cité à la question 8. peut être amélioré par un tri dit en place. Pour chaque algorithme de tri A, B, et C, dire si un tri en place améliore la performance du tri.

## Tri à bulles

L'algorithme de tri à bulles reprend l'idée suivante : on place en position  $n-1$  le maximum de la liste  $L = L[ : n]$ , puis en position  $n-2$  le maximum de la sous-liste  $L[ : n-1]$ , et ainsi de suite jusqu'à placer en position 1 le maximum de la sous-liste  $L[ : 2]$ . Dès que la liste est totalement triée on arrête le calcul.

14. Écrire le code d'une fonction `remonter(L, i)` qui remonte le maximum de la sous-liste  $L[ : i + 1]$  jusqu'à la position  $i$ ; on procédera pour cela à des échanges successifs éventuels des contenus des cases 0 et 1, 1 et 2,  $\dots$ ,  $i-1$  et  $i$ , comme si une bulle remontait le long de la liste. Cette fonction renverra un booléen égal à `True` si aucun échange n'a été fait (i.e. si la sous-liste  $L[ : i + 1]$  était croissante) et à `False` sinon.
15. En déduire le code d'une fonction `tri_bulles(L)` qui trie la liste  $L$ , en arrêtant le calcul dès que la liste est triée.

16. Étudiez les performances de cet algorithme de tri.
17. Cet algorithme de tri à bulle est-il analogue aux tris A, B ou C ?

## Recherche de la médiane

18. Expliquez comment on peut utiliser simplement un algorithme de tri pour rechercher la médiane d'une liste. Écrire une fonction `mediane_simple(L)` qui utilise la `fonction_3(L)` de la question 2. pour donner la médiane de L.
19. On peut aussi modifier directement les fonctions de la question 2. pour rechercher directement la médiane. On effectue les modifications suivantes. Expliquez pourquoi cet algorithme recherche la médiane et expliquez l'intérêt de ces modifications.

```

1  def fonction_1(L,debut,fin):
2      p = L[debut]
3      j = debut
4      for i in range(debut,fin):
5          if p > L[i]:
6              L = L[:j]+[L[i]]+L[j:i]+L[i+1:]
7              j +=1
8      return j,L
9
10 def fonction_2(L,debut,fin):
11     if debut+1>=fin:
12         return L[debut]
13     else:
14         j,L = fonction_1(L,debut,fin)
15         if j == len(L)//2:
16             m = L[j]
17             return m
18         elif j < len(L)//2:
19             debut_1 = j+1
20             fin_1 = fin
21             m = fonction_2(L,debut_1,fin_1)
22             return m
23         elif j > len(L)//2:
24             debut_2 = debut
25             fin_2 = j
26             m = fonction_2(L,debut_2,fin_2)
27             return m
28
29 def fonction_3(L):
30     m = fonction_2(L,0,len(L))
31     return m
32

```

20. Est-il possible de faire le même type de modification pour les algorithmes de tri B et C ? Justifier pour chaque algorithme de tri.

## Un autre algorithme de tri.

On considère M un entier strictement positif et L une liste d'entiers compris entre 0 et M-1.

21. Écrire une fonction comptage qui renvoie une liste L1 de longueur M telle que L1[i] soit égal au nombre d'éléments de L égaux à i.
22. Utiliser la fonction précédente pour écrire une fonction permettant de trier L.

23. Quelles sont les performances de cette fonction de tri ? Que remarquez vous par rapport aux tris A, B, et C ? Comment expliquer ce résultat ?
24. On vous donne sous forme d'une liste La Supplication de Svetlana Alexievitch. Les éléments de la liste sont les différents mots contenus dans le livre. En vous servant entre autre d'un algorithme de tri, imaginez un algorithme qui permettra de vérifier rapidement si les citations données par les futurs étudiants se trouvent dans l'œuvre ou pas. Détaillez le principe de votre algorithme. Puis écrivez le programme python correspondant.