

Algorithmes de tri

Un algorithme de tri est, en informatique ou en mathématiques, un algorithme qui permet d'organiser une collection d'objets selon un ordre déterminé (Référence [wikipedia](#)).

Les tris sont intéressants du point de vue de l'apprentissage de l'algorithmique.

Tri à bulles

-  [Tri à bulles](#)

Adapter en Python le pseudo code proposé. La représentation du déroulement du tri est également intéressante à développer.

Quicksort

Version récursive de l'algorithme, avec un pivot arbitraire.

[quicksort_01.py](#)

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
Code compact de la fonction quicksort. Exemple de tri d'éléments
entiers.
"""

def quicksort(li):
    if li == []: return []
    pivot = li[0]
    equal = [pivot]
    lesser = quicksort([x for x in li[1:] if x <= pivot]) # si x <
pivot : élimination des doublons
    greater = quicksort([x for x in li[1:] if x > pivot])
    return lesser + equal + greater

from numpy import random
a=random.randint(0,1000,10)
print(a,len(a))
b=quicksort(a)
print(b,len(b))
```

On peut aussi rendre la fonction encore plus compacte, mais moins lisible :

```
...
def quicksort(li):
    if li == []: return []
    pivot = li[0]
    return quicksort([x for x in li[1:] if x <= pivot]) + [pivot] +
quicksort([x for x in li[1:] if x > pivot])
...
```

Variante avec choix aléatoire du pivot (en fait de son index dans la numérotation de la liste) :

```
...
    pivot_index = randrange(len(li))
    pivot = li[pivot_index]
...
```

Références :

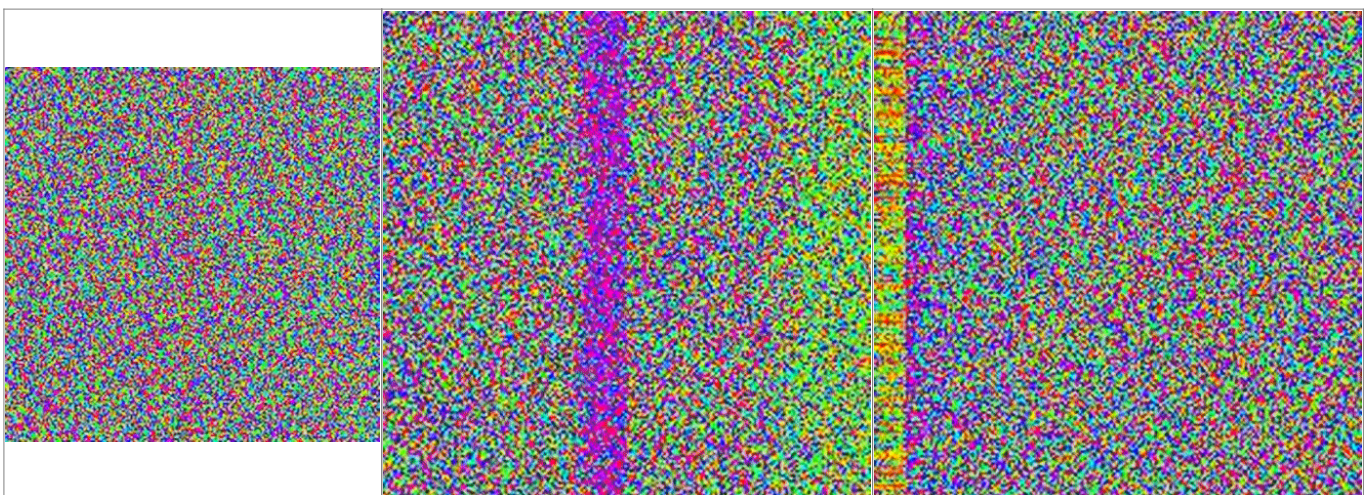
- <http://stackoverflow.com/questions/18262306/quick-sort-with-python>
- [Wikipedia en](#) et [Wikipedia fr](#)

Comparaison des tris

L'efficacité des tris peut être comparée suivant la configuration des données initiales (avant tri) et leur nombre. Voir par exemple sur le site www.sorting-algorithms.com, ou cette vidéo :

[CZWNISuWkAEwE3q.mp4](#)

Cf. aussi cette autre visualisation : [Sorting Algorithms Visualized in Python](#), Using Python 3 and Scikit-Image



Algorithmes inefficaces

- [Tri stupide](#) (bogo sort)

Références

- Sources de codes (qualité à vérifier)
<https://github.com/thecodershub/algorithms/tree/master/python/sort>
- [Sorting Algorithms in Python](#) Santiago Valdarrama, Real Python

From:

<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:

<https://dvillers.umons.ac.be/wiki/teaching:progappchim:tris>

Last update: **2022/12/09 14:25**

