



TP - Tri d'une liste de nombres

Introduction

Ce TP a pour objectif d'implémenter différents algorithmes de tri d'une liste de nombres.

Vous pouvez utiliser cet exemple pour tester votre algorithme

```
non_trie = [4, 42, 13, 26, 27, 16, 32, 37, 43, 17, 49, 6, 25, 5, 38, 48, 35, 40, 7, 31, 10, 18, 41, 21, 19, 28, 30, 12, 33, 36, 47, 24, 29, 2, 0, 9, 15, 11, 14, 3, 23, 20, 8, 44, 39, 34, 46, 45, 22, 1]
trie = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]

assert tri(non_trie) == trie
```

1 - Bubble sort

Cet algorithme a pour principe de faire plusieurs “passes” sur une liste de nombres, en réalisant des échanges

Chaque paire de nombre est comparée, et échangée afin qu’elle soit ordonnée par ordre croissant.

Example de passe:

5 2 9 1 4

2 < 5, on inverse le 5 et le 2

2 5 9 1 4

9 > 5, on ne fait rien

1 < 9, on inverse le 9 et le 1

2 5 1 9 4

4 < 9, on inverse le 9 et le 4

2 5 1 4 9

Lors de cette passe d’exemple, 3 échanges ont été faits

L’algorithme continue de faire ces passes en boucle jusqu’à ce qu’aucun échange n’est fait sur une passe. Alors, on considère l’algorithme terminé, et la liste est triée

Implémenter cet algorithme et valider son fonctionnement

2 - Fusion sort

Le tri par fusion consiste à diviser la liste entière de nombres en plusieurs sous-listes.

On trie la sous-liste avec un autre algorithme, ou en appelant récursivement l'algorithme de tri par fusion

On ajoutera au résultat le plus petit des nombres apparaissant en premier dans la liste

Example:

```
[ 1, 5, 6 ] [ 2, 3, 4 ]  
[]
```

On ajoute "1"

```
[ 5, 6 ] [ 2, 3, 4 ]  
[ 1 ]
```

On ajoute "2"

```
[ 5, 6 ] [ 3, 4 ]  
[ 1, 2 ]
```

On ajoute "3"

etc...

Implémenter un algorithme de fusion qui subdivise en 5 sous-listes, et valider son fonctionnement