

ACT ALGORITHME DE DICHOTOMIE

Voir l'animation :

<https://pixees.fr/tranche-de-formation-toi-meme-chapitre-3-la-dichotomie-et-les-prenoms/>

I. Rappels de tris d'une liste

Il existe deux façons de trier une liste en Python.

1. La fonction **sorted()**.

Cette fonction trie la liste et la renvoie. La liste de départ n'est pas modifiée et il faut stocker le résultat dans une autre variable si on doit l'utiliser.

2. La méthode **.sort()**.

Cette méthode trie la liste « en place ». Celle-ci prend la valeur de la liste triée, et la liste de départ est perdue.

Exemples :

```
>>L = [1, 2, 6, 5, 8, 3, 10]
>>L_triee = sorted(L)
>>L
>>L_triee
```

```
>>L = [1, 2, 6, 5, 8, 3, 10]
>>L.sort()
>>L
```

II. Principe de l'algorithme de dichotomie

1. Ecriture de l'algorithme

L'algorithme de dichotomie s'applique à un **tableau trié**.

A chaque étape,

- Diviser le tableau en deux parties égales.
- Prendre la valeur du milieu et comparer avec la valeur cherchée.
- Supprimer la partie du tableau dans laquelle ne se trouve pas la valeur cherchée.

Travail demandé :

- 1) Exécuter cet algorithme pas à pas, « à la main », sur la liste [1 , 5 , 12 , 15 , 18 , 24 , 54] dans laquelle on cherche la valeur 18. Recommencer avec la valeur 7.
- 2) Ecrire en langage naturel, ou pseudo-code, un algorithme de dichotomie permettant de déterminer si un nombre choisi est dans une liste triée de nombres entiers de longueur n.
- 3) Coder cet algorithme en Python et le tester pour différentes valeurs.

2. Correction de l'algorithme

- 4) Montrer que la propriété suivante est bien un invariant de boucle :
Si v est présent dans tab , alors il existe a et b tels que $tab[a] \leq v < tab[b]$
- 5) Prouver que l'algorithme se termine après un nombre fini d'étapes. On peut utiliser la variante de boucle $d - g$ où d et g sont les indices des bornes de la liste.

3. Coût de l'algorithme

On reprend l'exemple précédent.

- 6) Ecrire un autre programme qui effectue la même tâche mais en parcourant cette fois la liste du premier au dernier élément. Quel est son coût ?
- 7) Comparer leur temps d'exécution pour différentes valeurs de n et regrouper les résultats dans un tableau :

n =	Coût Dichotomie Algo question 3)	Coût standard Algo question 6)
10		
100		
1000		
10000		
...

- 8) Faire afficher la courbe à l'aide de **matplotlib**. Le coût est-il linéaire ?
- 9) A partir des fonctions proposées, déterminer la complexité de l'algorithme de dichotomie proposé.