

# Recherche dichotomique dans un tableau trié

## Activité débranchée

### Question 1

**Est-ce que ton prénom est dans la liste ci-dessous ?**

(attention, tu n'as que 3 secondes pour répondre...)

Naïcha	Lisa	Safia	Noura	Cherryane	Sukenya	Astrid	Islam
Mila	Majd	Alexis	Younes	Elyas	Noemie	Liza-Marie	Yoriss
Juliette	Ines	Aymene	Aboubacar	Ahmed	Amar	Oussema	Thilou
Enzo	Nasra	Mounira	Issafidine	Zohra	Kadidja	Nasrine	Kilian
Sophie	Nadia	Camille	Canthin	Nancy	Stella	Mariama	Yenni
Sérine	Chérine	Senouci	Kenza	Jenna	Mailys	Thea	Tanlimie
Morgane	Iliana	Nesrine	Alyssia	Chloe	My-Lan	Gilbert	Pablo
David	Axel	Naim	Rana	Mireille	Dounia	Aytac	Alaina
Rayan	Bouchra	Ambrine	Mihidhwari	Sonia	Sylvain	Célia	Amine
Zelal	Maéva	Zineb	Raphael	Kais	Shaïna	Ouiem	Malcolm
Malha	Marwan	Kassi	Ryad	Djasmina	Imen	Tom	Yanis
Sofia	Elarif	Denis	Benoit	Naem	Estelle	Linas	Jybril
Marion	Havin	Chaïma	Ayman	Adlen	Weam	Emma	Adam
Maïssane	Nawwâs	Elyne	Abdelhakim	Pascal	Joris	Mouzdalifah	Sarah
Rania	Djena	Hedi	Anfaïta	Theo	Lucas	Syrine	Wail
Nouh	Yacine	Siloe	Linda	Louna	Loris	Fatima	Salim
Olgaline	Nedjma	Isaac	Mohamed	Anass	Dana	Marilyne	Wael

Trouvé?...

### Question 1 (la suite...)

Bon, même question mais cette fois-ci les prénoms sont *triés par ordre alphabétique*.

**Est-ce que ton prénom est dans la liste ci-dessous ?**

(attention, tu n'as que 3 secondes pour répondre...)

Abdelhakim	Aytac	Elyas	Juliette	Mariama	Naim	Rania	Sérine
Aboubacar	Benoit	Elyne	Jybril	Marilyne	Nancy	Raphael	Tanlimie
Adam	Bouchra	Emma	Kadidja	Marion	Nasra	Rayan	Thea
Adlen	Camille	Enzo	Kais	Marwan	Nasrine	Ryad	Theo
Ahmed	Canthin	Estelle	Kassi	Maéva	Nawwâs	Safia	Thilou
Alaina	Chaïma	Fatima	Kenza	Mailys	Naïcha	Salim	Tom
Alexis	Cherryane	Gilbert	Kilian	Maïssane	Nedjma	Sarah	Wael
Alyssia	Chloe	Havin	Linas	Mihidhwari	Nesrine	Senouci	Wail
Amar	Chérine	Hedi	Linda	Mila	Noemie	Shaïna	Weam
Ambrine	Célia	Iliana	Lisa	Mireille	Nouh	Siloe	Yacine
Amine	Dana	Imen	Liza-Marie	Mohamed	Noura	Sofia	Yanis
Anass	David	Ines	Loris	Morgane	Olgaline	Sonia	Yenni
Anfaïta	Denis	Isaac	Louna	Mounira	Ouiem	Sophie	Yoriss
Astrid	Djasmina	Islam	Lucas	Mouzdalifah	Oussems	Stella	Younes
Axel	Djena	Issafidine	Majd	My-Lan	Pablo	Sukenya	Zelal
Ayman	Dounia	Jenna	Malcolm	Nadia	Pascal	Sylvain	Zineb
Aymene	Elarif	Joris	Malha	Naem	Rana	Syrine	Zohra

Des remarques ?

## Question 2

Par groupe de 2, **proposer** au moins 2 algorithmes possibles pour chercher un prénom parmi 136 possibles et déterminer pour chacun le nombre maximal d'opérations.

## Recherche dichotomique

La recherche dichotomique est un procédé classique de recherche **très efficace**.

Pour fonctionner, il faut un tableau **trié**, par exemple par ordre croissant.

Tableau trié



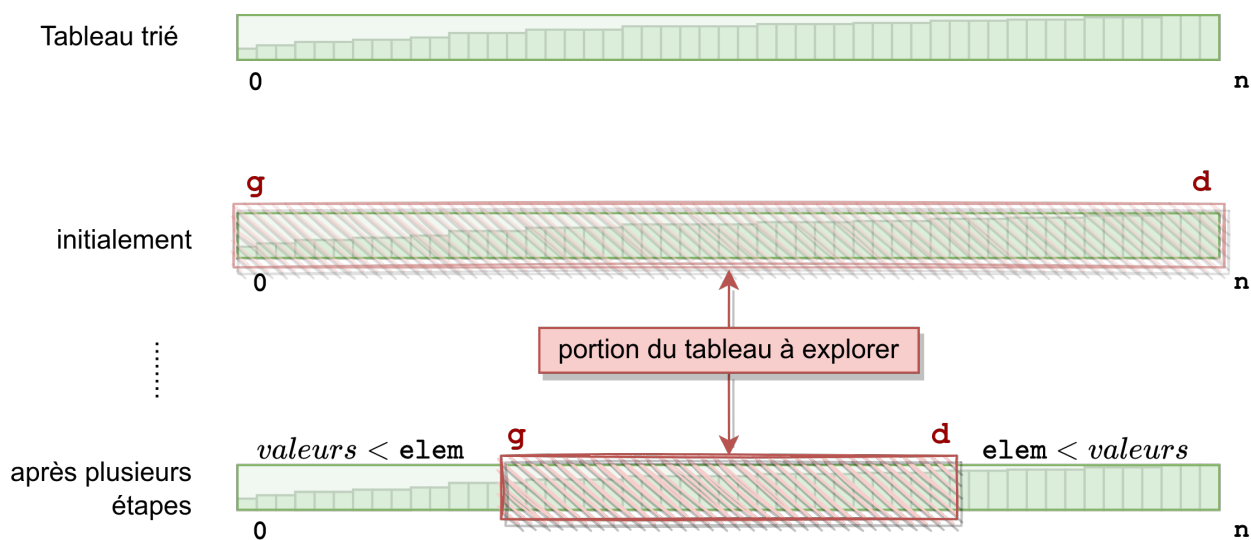
Supposons que l'on cherche l'élément  $elem$  dans le tableau  $tab$  de taille  $n$ .

Pour mettre en œuvre la recherche dichotomique, on va délimiter la **portion** du

tableau dans laquelle la recherche est actuellement réduite à l'aide de deux **indices** :  $g$  et  $d$ .

Initialement, la recherche se fait dans le tableau en entier et donc on prendra pour commencer :

- $g \leftarrow 0$
- $d \leftarrow n - 1$



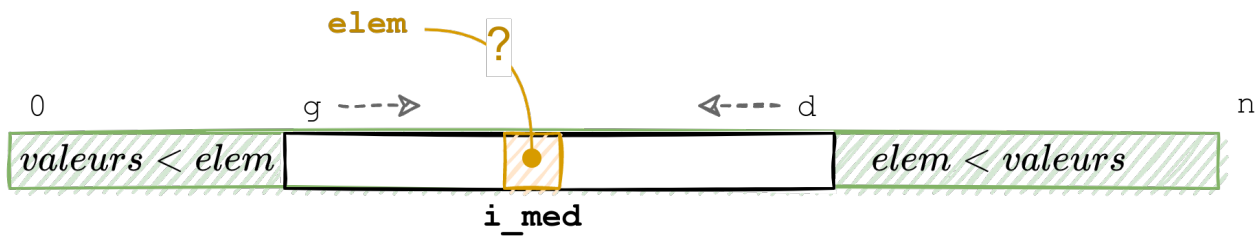
Au fur et à mesure de la recherche, la portion de tableau à explorer va se réduire. Les indices  $g$  et  $d$  seront modifiés pour que si  $elem$  est dans le tableau, il soit forcément entre les indices  $g$  et  $d$ .

$elem$  ne peut se trouver que dans  $tab[g \dots d]$ .

Ainsi,

- tout élément d'indice inférieur à  $g$  est inférieur à  $elem$
- tout élément d'indice supérieur à  $d$  est supérieur à  $elem$ .

Supposons maintenant la situation courante du schéma :



La dichotomie se poursuivra **tant que** la portion de recherche n'est pas vide et à chaque tour de boucle, il faudra comparer `elem` avec la valeur médiane de la portion `tab[i_med]`.

- Si cette valeur médiane est égale à `elem` alors on renvoie l'indice de cette valeur (le renvoie arrête instantanément la fonction de recherche).
- Sinon, on met à jour la portion de recherche en modifiant
  - soit `g` : pour que la prochaine boucle explore la portion **supérieure** à la valeur médiane
  - soit `d` : pour que la prochaine boucle explore la portion **inférieure** à la valeur médiane.

### Question 3

Intéressons-nous à la valeur médiane de la portion du tableau. **Proposer** une valeur de `i_med` en fonction de `g` et de `d`.

### Question 4

Pour vérifier que tu as compris l'algorithme, **valide** l'activité 4 sur Moodle :



## 04. Activité - dichotomie à la main

### Question 5

Pour définir correctement la boucle **tant que** :

1. **Déterminer** sa condition d'arrêt.
2. **Déterminer** ses conditions initiales.

### Question 6

**Implémenter** la recherche dichotomique.

Pour cela, compléter la fonction `recherche_dichotomique()` qui prend en argument un tableau trié `tab` et une valeur `elem`.

La fonction doit renvoyer l'indice de `elem` s'il est présent dans `tab` et `-1` sinon.

```
[ ]: def recherche_dichotomique(tab: list, elem) -> int:
    """
    Recherche dichotomique dans un tableau trié
    """
    #
    ...
    # todo
    ...
```

## Analyse

Nous allons dans cette partie analyser l'**efficacité** de l'algorithme de recherche dichotomique.

Pour cela, nous allons le comparer avec l'algorithme de *recherche brute*, méthode naïve qui se contente de parcourir le tableau du début à la fin tant que la valeur cherchée n'est pas présente.

### Question 7

**Lire** le paragraphe suivant et **indiquer** ce qu'est le *pire des cas* pour les deux algorithmes de recherche.

Pour mesurer l'efficacité d'un algorithme, une méthode classique est de compter le nombre d'opérations effectuées par l'algorithme *dans le pire des cas*.

Nous n'allons pas compter le nombre d'opérations, mais effectuer une mesure quantitative qui y est fortement liée. Nous allons mesurer la durée de chacun des algorithmes sur des grands tableaux.

### Question 8

**Implémenter** la fonction `recherche_brute` qui prend en argument un tableau trié `tab` et une valeur `elem`.

L'algorithme de cette fonction parcourt toutes les cases du tableau `tab` jusqu'à trouver une valeur égale à `elem`. Cette fonction renvoie `-1` si la recherche est infructueuse et l'indice de `elem` sinon.

```
[ ]: def recherche_brute(tab: list, elem) -> int:
    """
    Algorithme de recherche brute dans un tableau trié
    """
    ...
    # todo
    ...
```

## Question 9

**Compléter** le code ci-dessous afin de mesurer la durée de chacune des fonctions de recherche avec des tableaux de différentes tailles.

```
[ ]: from time import time  
...  
...
```

## Pour aller plus loin

### Question 10

Tu connais le jeu *Devine Nombre*. Il faut y deviner un nombre en ayant comme réponses à tes tentatives : *c'est plus petit* ou *c'est plus grand*.

1. Écris un algorithme peu efficace pour trouver ce nombre.
2. Écris un algorithme plus efficace pour trouver ce nombre.