

## - Travail à faire pour la prochaine séance en demi groupe -

Lundi 26 avril à 16 heures : groupe 2

Lundi 3 mai à 16 heures : groupe 1

Lien visio: <https://lycee.cned.fr/cv/269718/221>

(inutile de passer par Atrium puis Pronote pour utiliser ce lien)

### Documents joints :

- fichier `pnssi_algos_tris_et_recherches.pdf` contenant les explications des algorithmes.
- fichier `pnssi_programmes_bases.py` contenant les algorithmes du programme.

### Objectifs :

Assimiler l'algorithme de recherche dichotomique dans une liste triée, pages 8 et 9.

Question 1 : Bien lire les pages 8 et 9 du fichier pdf et noter vos questions.

### Questions 2 :

Modifier le programme principal du **fichier python** pour **tester les deux fonctions de tri** par exemple, **tester** différentes **tailles** de tableaux et de **valeurs** à trouver, les petites listes étant plus visuelles pour l'affichage, que vous pouvez également modifier à votre convenance.

Il peut également être intéressant d'insérer une ligne d'affichage dans la fonction `recherche_dicho(tableau, valeur)`.

Cela permet de visualiser le fonctionnement de l'algorithme étapes par étapes dans la console:

- **Insérer** une ligne dans la fonction `recherche_dicho()` afin d'afficher au début de chaque boucle la liste dans laquelle la recherche est effectuée.
- **Tester** dans la console.
- Avez-vous des **observations** à faire sur les intervalles de recherche qui s'affichent ?

Questions 3 : Faire fonctionner l'algorithme dans l'exemple suivant, puis **compléter** le tableau.

Recherche de **valeur = 8** dans la liste triée **Tab = [1, 2, 3, 4, 5, 6, 7, 8]**

numéro étape	indice_milieu	indice_mini	indice_maxi	renvoie fonction
0	x	0	7	x
1	3	...	...	...
2	...	...	...	...

- Combien d'étapes seraient nécessaires pour rechercher la valeur **10** dans **Tab** ?

Questions 4 :

- **Insérer** dans la fonction un **compteur** du nombre d'étapes, c'est-à-dire du **nombre de boucles effectuées**. La fonction devra ainsi renvoyer non plus une seule valeur entière, mais un **tuple** d'entiers, comme dans la fonction `extremums()` expliquée page 4 du pdf.
- **Tester** dans la console sur des listes de tailles différentes, afin de pouvoir **compléter** le tableau suivant :

taille $n$ de la liste	8	16	32	64	128	256	512	1024	2000	5000
nombre d'étapes maximum										

Question 5 :

On souhaite programmer une fonction qui prend un seul argument en entrée, une longueur  $n$ . Cette fonction doit renvoyer le nombre maximum d'étapes nécessaires à la recherche de la présence d'une valeur quelconque dans une liste triée de longueur  $n$ .

- Pour être certain d'obtenir la bonne réponse, **quelle liste et quelle valeur à trouver** peut-on choisir ?