

Exercice : Recherche par dichotomie

On redonne l'algorithme de recherche par dichotomie en langage python.

```
a = 0
b = len(t) - 1
while a <= b :
    m = (a + b) // 2
    if v < t[m] :
        b = m - 1
    elif v > t[m] :
        a = m + 1
    sinon return m
return -1
```

La valeur -1 est retournée lorsque le nombre n'est pas présent dans le tableau. On pourrait choisir une autre valeur ou même une chaîne de caractère !

Exercice 1

Max et Lilly joue à devine nombre. Max pense à un nombre entier compris entre 1 et 100 et Lilly doit le trouver. À chaque nombre proposé par Lilly, Max lui dit s'il est plus petit ou plus grand.

Si Lilly applique la recherche par dichotomie, en combien d'essais au maximum peut-elle trouver le nombre de Max ?

Exercice 2

Soit T un tableau trié tel que : $T = [8, 8, 17, 21, 23, 27, 28, 45, 57, 71, 77, 84, 88, 95, 97]$.

- 1) On veut vérifier la présence du nombre 23 dans le tableau.
 - a) Retracer toutes les étapes de l'algorithme de recherche par dichotomie.
 - b) Combien d'itérations ont été nécessaires ?
- 2) Faire de même avec le nombre 88.
- 3) Faire de même avec le nombre 75.

Exercice 3

- 1) Créer un tableau T de valeurs choisies aléatoirement entre 1 et 1000.
- 2) Trier ce tableau avec une des fonctions de tri de python.
- 3) Écrire l'algorithme de recherche par dichotomie en langage python.
- 4) Tester votre programme avec la recherche de différentes valeurs du tableau et des valeurs qui ne sont pas dans le tableau.
- 5) Pour connaître le nombre maximum d'itérations, on peut utiliser la fonction mathématique $\log_2(n)$ où n est la dimension du tableau. On arrondira à l'entier supérieur.
 - a) Importer le module **math** et calculer la valeur $\log_2(n)$:

```
import math
Imax = math.log2(20)
```

- b) Quel est le nombre maximal d'itérations de votre recherche dichotomique ?
- 6) Écrire une boucle qui donne toutes les recherches dichotomiques pour trouver tous les nombres de votre tableau et le nombre d'itérations dans chaque cas.