

Recherche dichotomique

Fiche de synthèse

① RECHERCHE SÉQUENTIELLE

```
def chercher(liste, x):
    for element in liste:
        if element == x:
            return True
    return False
```

La **recherche séquentielle** compare les éléments un par un. Elle fonctionne même si la liste n'est pas triée, mais, dans le pire cas, élément absent ou en dernière position, elle examine les n éléments : sa complexité est en $O(n)$.

② LE PRINCIPE DE LA DICHOTOMIE

Dans un dictionnaire, on ouvre vers le milieu, puis on continue dans la moitié où se trouve le mot. Pour deviner un nombre entre 1 et 100, on propose toujours le milieu de l'intervalle restant : 73 est trouvé en six essais, 50, 75, 62, 68, 71, puis 73.

Définition 1 (Recherche dichotomique) Dans une liste triée, la **recherche dichotomique** compare l'élément cherché à celui du milieu de la zone de recherche, puis ne garde que la moitié qui peut le contenir. À chaque étape, la zone est divisée par deux.

```
def recherche_dichotomique(liste, element):
    debut = 0
    fin = len(liste) - 1
    while debut <= fin:
        centre = (debut + fin) // 2
        if liste[centre] == element:
            return True
        elif liste[centre] < element:
            debut = centre + 1
        else:
            fin = centre - 1
    return False
```

Pour renvoyer la position de l'élément, on écrit `return centre` au lieu de `return True`, et `return None` à la fin s'il est absent.

③ EXEMPLE

On cherche 14 dans [1, 2, 5, 9, 10, 14, 17, 24, 41].

Étape	debut	fin	centre	Comparaison
1	0	8	4	10 < 14 : à droite
2	5	8	6	17 > 14 : à gauche
3	5	5	5	14 = 14 : trouvé

Pour 7, absent, quatre étapes mènent à `debut = 3` et `fin = 2` : la condition `debut <= fin` est fausse, la fonction renvoie `False`.

④ COMPLEXITÉ

Après k étapes, il reste au plus $n/2^k$ éléments : il faut au plus $\lfloor \log_2 n \rfloor + 1$ étapes. $\log_2 n$ est l'exposant k tel que $2^k = n$; $\lfloor x \rfloor$ est la partie entière de x . La complexité est en $O(\log n)$, **logarithmique**.

n	8	16	64	128	10^6
Séquentielle	8	16	64	128	10^6
Dichotomie	4	5	7	8	20

⑤ TERMINAISON ET CONDITION

Le **variant** `fin - debut + 1`, le nombre d'éléments restant à examiner, est un entier positif qui diminue d'au moins 1 à chaque tour : la boucle se termine.

⚠ Attention. La recherche dichotomique exige une liste **triée** : sinon, elle peut déclarer absent un élément présent. On trie une fois, puis chaque recherche est très rapide.

📌 À retenir.

- La recherche séquentielle est en $O(n)$, et fonctionne sur toute liste.
- La dichotomie divise la zone par deux à chaque étape : $O(\log n)$, 20 comparaisons pour un million d'éléments.
- Elle exige une liste triée.