Recherche dichotomique

Recherche dichotomique

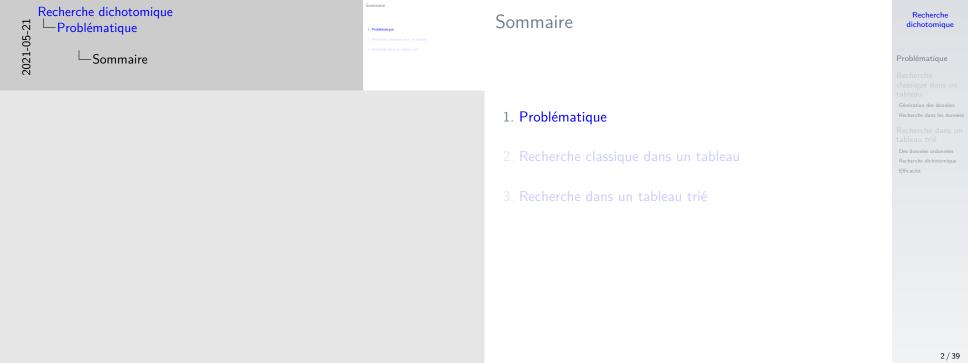
Christophe Viroulaud

Recherche dichotomique

Christophe Viroulaud

Première - NSI

Recherche dichotomique



Problématique

Problématique

Rechercher un élément dans un tableau est une opération courante. Cette tâche a un coût qui dépend de la taille du tableau. Cependant, si le tableau est déià trié il est possible d'accélérer grandement la recherche.

ment implémenter une recherche efficace dans un tableau trié :

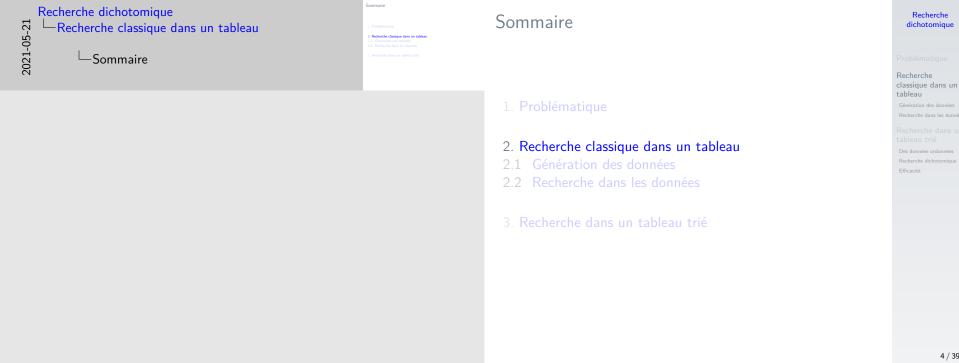
Rechercher un élément dans un tableau est une opération courante. Cette tâche a un coût qui dépend de la taille du tableau. Cependant, si le tableau est déjà trié il est possible d'accélérer grandement la recherche.

Problématique

Recherche

dichotomique

Comment implémenter une recherche efficace dans un tableau trié?



4/39

Recherche dichotomique

Recherche classique dans un tableau
Génération des données
Génération des données

Génération des données

On demande à dix mille personnes de penser à un nombre entre 0 et 1000000 et on stocke les résultats dans un tableau au fur et à mesure des réponses.

de dix mille entiers compris entre 0 et 1000000.

Génération des données

On demande à dix mille personnes de penser à un nombre entre 0 et 1000000 et on stocke les résultats dans un tableau au fur et à mesure des réponses.

Activité 1 : Construire par compréhension un tableau de dix mille entiers compris entre 0 et 1000000.

Recherche dichotomique

Problèmatique

classique dans un ableau

Génération des données

Recherche dans les données

Des données ordonnées Decherche dichotomique

Correction

1 entiers = [randint(0, 1000000) for _ in range(100000)]

Jeu de données

dichotomique

Recherche

Problématique

Recherche classique dans ur

Génération des données

Recherche dans les données

cherche dans un pleau trié

données ordonnées erche dichotomique acité



Sommaire

- 1. Problématique
- 2. Recherche classique dans un tableau
- 2.1 Génération des données
- 2.2 Recherche dans les données
- 3. Recherche dans un tableau tri

herche

Recherche

dichotomique

herche sique dans un

Recherche dans les données

cherche dans ur leau trié données ordonnées therche dichotomique

7 / 39



Recherche dans les données

Pour vérifier la présence d'une valeur dans les données, il faut parcourir le tableau élément par élément.

3	180	1007	56		2178	8	
---	-----	------	----	--	------	---	--

FIGURE 1 – Parcours séquentiel

Recherche dichotomique

Problématique

lassique dans un ableau

Recherche dans les données

oleau trié s données ordonnées

cherche dichotomique



cas où l'élément n'est pas présent

À retenir

Dans le pire des cas la complexité temporelle de la recherche dépend du nombre d'éléments.

O(n)

Recherche dichotomique

Recherche dans les données

Recherche dichotomique -Recherche classique dans un tableau -Recherche dans les données

Activité 2 :

- - L Écrire la fonction recherche_classique(tab: list. cherche: int) → bool qui renvoie
 - True si l'entier cherche est présent dans le
 - Tester la fonction : vérifier si le nombre 575000 a
 - À l'aide de la méthode time de la bibliothèque time mesurer la dunée d'exécution de la fonction

Activité 2 :

- 1. Écrire la fonction recherche_classique(tab: list, cherche: int) \rightarrow bool qui renvoie True si l'entier cherche est présent dans le tableau.
- 2. Tester la fonction : vérifier si le nombre 575000 a été choisi par une personne.
- 3. À l'aide de la méthode time de la bibliothèque time mesurer la durée d'exécution de la fonction.

Recherche dichotomique

Recherche dans les données

Correction

def recherche_classique(tab: list, cherche: int) -> bool:
 """
Renvoie True si 'cherche' est dans 'tab'
"""
for element in tab:
 if element == cherche:
 return True
 # à la fin de la boucle on n'a pas trouvé 'cherche'
 return False

Recherche dichotomique

Problematique

classique dans un tableau

Génération des données

Recherche dans les données

ableau trié Des données ordonnées

Recherche dichotomique Efficacité Exemple de résultat

Correction

- deb = time()
- print(recherche_classique(entiers, 575000))
- fin = time()
- 4 print(fin—deb)

1 False

 $2 \quad 0.0066792964935302734$

Exemple de résultat

Recherche dichotomique

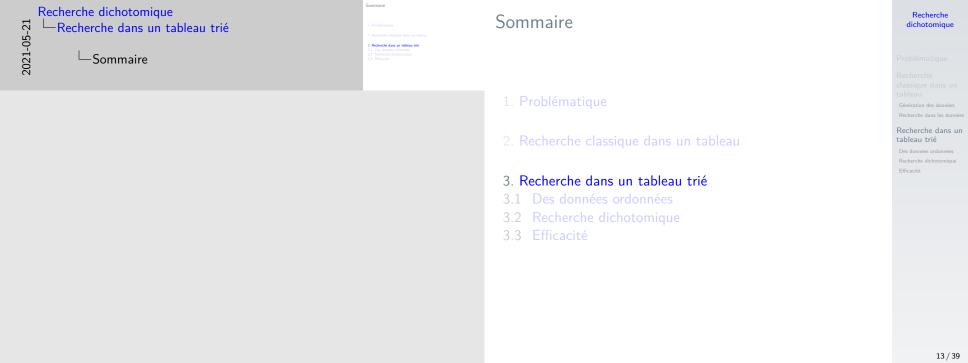
Problématique

echerche lassique dans un

Génération des données

Recherche dans les données

es données ordonnées



Recherche dichotomique

Recherche dans un tableau trié

Des données ordonnées

Des données ordonnées

Imaginone maintenant la mime expérience mais prenons la peine de trie les éléments au fur et à mesure de leur ajout dans la tableux de données.

3 8 56 180 1007 2178

PRICER 2 - Tableux trié

Des données ordonnées

Des données ordonnées

Imaginons maintenant la même expérience mais prenons la peine de trier les éléments au fur et à mesure de leur ajout dans le tableau de données.

3 8 56 180		1007	2178	
------------	--	------	------	--

FIGURE 2 – Tableau trié

Recherche dichotomique

Problématique

classique dans un tableau

Génération des données Recherche dans les données

tableau trié

Des données ordonnées

cherche dichotomique icacité

Activité 3 : Pour simplifier nous allons utiliser la méthode **sort** pour trier les données.

- 1. Construire par compréhension un tableau de dix mille entiers compris entre 0 et 1000000.
- 2. Trier le tableau.

Recherche dichotomique

Problématique

lassique dans un

Génération des données Recherche dans les données

tableau trié

Des données ordonnées

Recherche dichotomique Efficacité

Correction

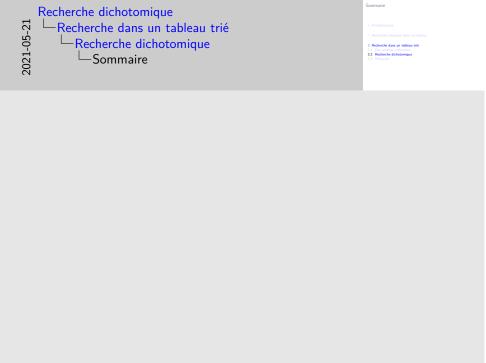
entiers = $[randint(0, 1000000) for _ in range(100000)]$ entiers.sort()

Jeu de données

Recherche

dichotomique

Des données ordonnées



Sommaire

- - ____

- 3. Recherche dans un tableau trié
- 3.1 Des données ordonnée
- 3.2 Recherche dichotomique
 - 3.3 Efficacité

Recherche

dichotomique

Recherche dichotomique

Les données étant triées, le principe de la dichotomie, pour charcher la présence d'un élément, consiste à

répéter l'opération jusqu'à trouver l'élément ou avoir

Recherche dichotomique

Les données étant triées, le principe de la dichotomie, pour chercher la présence d'un élément, consiste à :

- couper le tableau en deux parties égales,
- ▶ ne garder que la partie contenant l'élément,
- répéter l'opération jusqu'à trouver l'élément ou avoir une partie vide.

à peu près égales selon parité

Recherche dichotomique

Recherche dans un tableau trié
Recherche dichotomique

Cherchons 765 dans le tableau suivant :

3 8 56 180 256 302 765 1007 2

FROURE 3 - Séparona les dornées en deux parties

Cherchons 765 dans le tableau suivant :

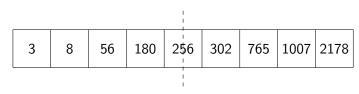


FIGURE 3 – Séparons les données en deux parties

Recherche dichotomique

Problématique

echerche assique dans un

Génération des données Recherche dans les données

bleau trié es données ordonnées

Recherche dichotomique Efficacité



FIGURE 4 – 256 n'est pas le nombre recherché et il est inférieur à 765

Recherche dichotomique

Problématique

echerche assique dans un

énération des données echerche dans les données

echerche dans un bleau trié es données ordonnées

Des données ordonnées Recherche dichotomique Efficacité Recherche dichotomique

Recherche dans un tableau trié
Recherche dichotomique



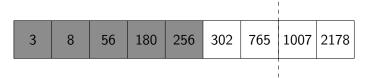


FIGURE 5 – Séparons les données restantes en deux parties

itique

Problematique

echerche assique dans un

Recherche

dichotomique

Sénération des données

echerche dans un bleau trié es données ordonnées

Des données ordonnées Recherche dichotomique Efficacité





FIGURE 6 – Nous pouvons éliminer la partie supérieure à 765

22 / 39

Recherche

dichotomique

Recherche dichotomique



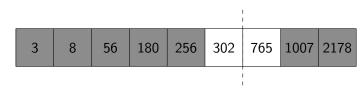


FIGURE 7 – Dernière séparation

Recherche dichotomique

Problématique

echerche assique dans un ableau

cherche dans les données

ableau trié

Des données ordonnées Recherche dichotomique

cacité





FIGURE 8 – 765 a été trouvée en trois itérations

Recherche

dichotomique

Recherche dichotomique

Recherche dichotomique

Recherche dans un tableau trié

-Recherche dichotomique

 En pratique, on utilise les indices pour trouver le milieu.

En pratique, on utilise les indices pour trouver le milieu.

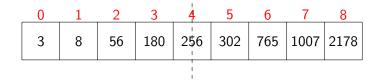


FIGURE 9 - $\frac{8+0}{2}$ = 4 l'indice médian est 4

- ► 256 n'est pas le nombre recherché,
- lest inférieur au nombre recherché.

Recherche dichotomique

Problématique

Recherche

tableau Génération des données

> ecnercne dans u ableau trié

Des données ordonnées Recherche dichotomique

Recherche dichotomique Recherche dans un tableau trié Recherche dichotomique



1. l'indice est un entier

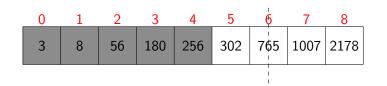


FIGURE
$$10 - \frac{8+5}{2} = 6$$
 l'indice médian est 6

- ▶ 765 est le nombre recherché,
- la recherche s'arrête.

Recherche dichotomique

Problématique

Recherche

tableau

Génération des données

echerche dans u

Des données ordonnées Recherche dichotomique Activité 4 : Écrire la fonction recherche_dicho (tab list, cherche: int)

bool qui applique le principe de la dichotomie. Pour séparer les données en deux parties (à peu près) égales il faudra calculer l'indice médian de la partie encore valide.

Activité 4 : Écrire la fonction recherche_dicho(tab: list, cherche: int) → bool qui applique le principe de la dichotomie. Pour séparer les données en deux parties (à peu près) égales il faudra calculer l'indice médian de la partie encore valide.

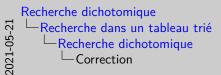
Recherche dichotomique

Problématique

classique dans un tableau

> cherche dans un bleau trié

Des données ordonnées
Recherche dichotomique
Efficacité



Correction

3

def recherche_dicho(tab: list, cherche: int) -> bool:

i_debut = 0 i_fin = len(tab)-1

```
def recherche_dicho(tab: list, cherche: int) -> bool:
   i\_debut = 0
   i_{fin} = len(tab) - 1
```

Recherche

dichotomique

Recherche dichotomique

while i_fin >= i_debut: i_milieu = (i_debut+i_fin) // 2

while i_fin >= i_debut: $i_milieu = (i_debut+i_fin) // 2$

Correction

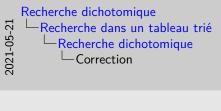
 $if \ cherche == tab[i_milieu] : \\$ return True

dichotomique

Recherche

Recherche dichotomique

30 / 39



```
Correction
```

elif cherche < tab[i_milieu]: i_fin = i_milieu-1 else: # cherche > tab[i_milieu]

```
elif cherche < tab[i_milieu] :</pre>
   i_fin = i_milieu-1
else: # cherche > tab[i_milieu]
   i\_debut = i\_milieu+1
```

Recherche

dichotomique

Recherche dichotomique

2021-05-21

Correction

```
def recherche_dicho(tab: list, cherche: int) -> bool:
       i debut = 0
       i_{fin} = len(tab) - 1
       while i fin >= i debut:
          i_milieu = (i_debut+i_fin) // 2
          if cherche == tab[i_milieu] :
              return True
          elif cherche < tab[i_milieu] :</pre>
             i fin = i milieu-1
          else: # cherche > tab[i_milieu]
10
              i debut = i milieu+1
       # à la fin de la boucle on n'a pas trouvé 'cherche'
12
       return False
13
```

Recherche dichotomique

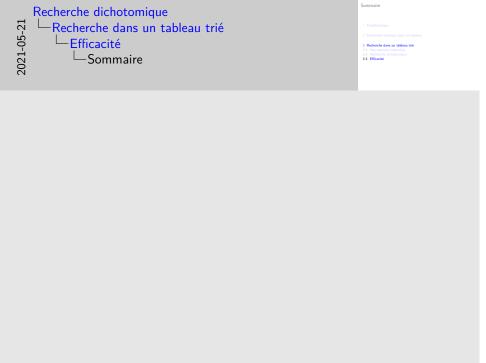
Problematique

Recherche classique dans un

énération des données

ableau trié Des données ordonnées

Recherche dichotomique Efficacité



Sommaire

- -

- 3. Recherche dans un tableau trié
- 3.1 Des données ordonnée
- 3.2 Recherche dichotomiq
- 3.3 Efficacité

Recherche

dichotomique

Efficacité

Efficacité

 $\frac{n}{2^{\times}} = 1$

 $\Leftrightarrow n = 2^x$

0 = pas trouvé

2021-05-21

À chaque itération la quantité de données (notée n) à étudier est divisée par deux. Dans le pire des cas, on divise jusqu'à ce que la taille de la partie restante soit inférieure ou égale à 1.

$$\frac{1}{2^x} = 1$$

$$\Leftrightarrow n = 2^x$$

$$n=2$$

Recherche

dichotomique

Efficacité

Activité 5 :

- Encadrer la valeur de x entre deux entiers, si le tableau contient n = 10000 éléments.
 Mesuurr la durée d'exécution de la fonction et la
- comparer à celle de la recherche classique.

Activité 5 :

- 1. Encadrer la valeur de x entre deux entiers, si le tableau contient n = 10000 éléments.
- 2. Mesurer la durée d'exécution de la fonction et la comparer à celle de la recherche classique.

Recherche dichotomique

Problématique

classique dans un tableau

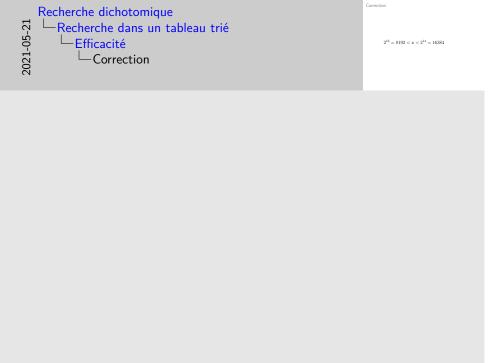
énération des données Recherche dans les données

Recherche dans un tableau trié

Des données ordonnées

Recherche dichotomique

Efficacité





 $2^{13} = 8192 < x < 2^{14} = 16384$

Recherche

dichotomique

Efficacité

facteur 10

2021-05-21

Correction

Correction

1 deb = time()
2 print(recherche_dicho(entiers, 575000))
3 fin = time()
4 print(fin-deb)

Exemple de résultat

False 0.0009853839874267578

- deb = time()
- print(recherche_dicho(entiers, 575000))
- fin = time()
- 4 print(fin—deb)

1 False

2 0.0009853839874267578

Exemple de résultat

Recherche dichotomique

Problématique

scherche Issique dans un

ération des données herche dans les données

cherche dans pleau trié s données ordonnées

Recherche dichotomique Efficacité

Recherche dichotomique Recherche dans un tableau trié Efficacité



$$\log_2 n = \frac{\ln n}{\ln 2}$$

À retenir

La complexité temporelle de la recherche dichotomique est :

$$\log_2 n = x$$

Recherche dichotomique

Problématique

ssique dans un bleau

> che dans les données erche dans un

Des données ordonnées Recherche dichotomique Efficacité

Code complet

Le code complet se trouve ici.

Recherche

dichotomique

Efficacité