Algorithmique

Recherche dichotomique

Yannick CHISTEL

Lycée Dumont d'Urville - CAEN

Mars 2020

Recherche dichotomique

Principe

La **dichotomie** est un mot d'origine greque qui signifie « diviser en deux ».
La recherche d'une valeur dans un tableau peut être facilitée si celui-ci est trié. Dans ce cas, on divise successivement le tableau en 2 jusqu'à atteindre la valeur cherchée.

Algorithme

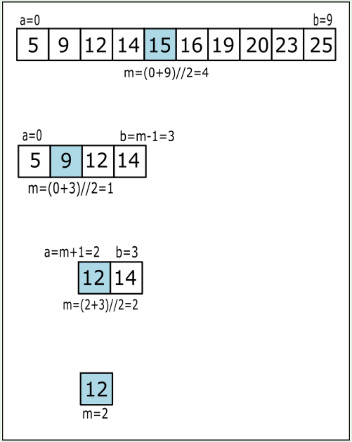
Voici une écriture de l'algorithme de recherche dichotomique dans un **tableau trié** :
- t désigne un tableau trié
- v est la valeur cherchée dans le tableau
- a, b et m sont les indices de position des valeurs dans le tableau.

$a \longleftarrow 1$ *...première valeur du tableau*
$b \longleftarrow$ longueur(t) *...dernière valeur du tableau*
tant que $a <= b$:
    $m \longleftarrow (a+b)//2$ *...m est la position au milieu*
    si $v < t[m]$ alors $b = m - 1$ *...v se trouve dans la première moitié*
    sinon si $v > t[m]$ alors $a = m + 1$ *...v se trouve dans la seconde moitié*
    sinon la valeur est trouvée en m
fin **tant que**
La valeur n'est pas trouvée

Exemple

Soit $T = [5, 9, 12, 14, 15, 16, 19, 20, 23, 25]$ un tableau trié. On cherche le nombre 12.

1. $a = 0$ et $b = 9$: $a < b$, donc on entre dans la boucle tant que.
2. On calcule la valeur de l'indice situé au milieu du tableau : $m = (0+9)/2 = 4,5$ donc $m = 4$. Comme $T[4] = 15 > 12$, alors le nombre cherché est positionné avant m, donc $b = m - 1 = 4 - 1 = 3$.
3. $a = 0 < b = 3$, donc on poursuit la boucle. On calcule la valeur de m : $m = (0+3)/2 = 1,5$ donc $m = 1$.
   Comme $T[1] = 9 < 12$, alors le nombre cherché est positionné après m, donc $a = m + 1 = 1 + 1 = 2$.
4. $a = 2 < b = 3$, donc on poursuit la boucle. On calcule $m = (2+3)/2 = 2,5$ donc $m = 2$. Comme $T[2] = 12$, le nombre est trouvé en position $m = 2$.



Exemple

Si le nombre n'est pas présent dans le tableau, il faut que la boucle se termine ! En voici les étapes avec la recherche du nombre 13.

1. $a = 0$ et $b = 9$: $a < b$, donc on entre dans la boucle tant que.
2. On calcule $m = (0 + 9)/2 = 4,5$ donc $m = 4$.
   Comme $T[4] = 15 > 13$, alors le nombre cherché est positionné avant m, donc $b = m - 1 = 4 - 1 = 3$.
3. $a = 0 < b = 3$, donc on poursuit la boucle : $m = (0 + 3)/2 = 1,5$ donc $m = 1$.
   Comme $T[1] = 9 < 13$, alors le nombre cherché est positionné après m, donc $a = m + 1 = 1 + 1 = 2$.
4. $a = 2 < b = 3$, donc on poursuit la boucle : $m = (2 + 3)/2 = 2,5$ donc $m = 2$.
   Comme $T[2] = 12 < 13$, alors le nombre cherché est positionné après m, donc $b = m - 1 = 3 - 1 = 2$.
5. $a = 2 = b$ donc la boucle se poursuit : $m = (2 + 2)/2 = 2$.
   $T[2] = 12 < 13$, alors le nombre cherché est positionné après m, donc $b = m - 1 = 2 - 1 = 1$.
6. $a = 2 > b = 1$, la boucle s'arête, aucun nombre n'a été trouvé.

Terminaison de l'algorithme

Variant de boucle

On appelle **variant de boucle** une quantité entière qui :

- doit être positive ou nulle pour rester dans la boucle ;
- décroit strictement à chaque itération

Si on trouve une telle quantité dans une boucle while, celle-ci se termine.

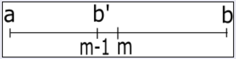
Preuve de la terminaison

Dans l'algorithme de recherche par dichotomie, le variant de boucle est $b - a$.

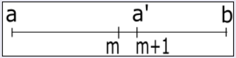
- Ce nombre est clairement supérieur ou égal à 0 puisque $a <= b$.
- Vérifions qu'elle décroît en distinguant 3 cas :

cas 1 : $t[m] == v$ alors on sort de la boucle.

cas 2 : $t[m] > v$ donc $b' - a < m - a < b - a$ donc la quantité $b - a$ décroit.



cas 3 : $t[m] < v$ donc $b - a' < b - m < b - a$ donc la quantité $b - a$ décroit.



$b - a$ est un variant de boucle positif qui décroit, assurant la terminaison de la **boucle while**.

Terminaison de l'algorithme

Exemple 1

On cherche le nombre 12 dans le tableau trié T. Les différentes valeurs de a et b sont :

1. $a = 0$ et $b = 9$ donc $b - a = 9 - 0 = \boxed{9}$
2. $a = 0$ et $b = 3$ donc $b - a = 3 - 0 = \boxed{3}$
3. $a = 2$ et $b = 3$ donc $b - a = 3 - 2 = \boxed{1}$

La quantité $b - a$ est décroissante et positive tout le temps de la boucle.

Exemple 2

On cherche le nombre 13 dans le tableau trié T. Les différentes valeurs de a et b sont :

1. $a = 0$ et $b = 9$ donc $b - a = 9 - 0 = \boxed{9}$
2. $a = 0$ et $b = 3$ donc $b - a = 3 - 0 = \boxed{3}$
3. $a = 2$ et $b = 3$ donc $b - a = 3 - 2 = \boxed{1}$
4. $a = 2$ et $b = 2$ donc $b - a = 2 - 2 = \boxed{0}$
5. $a = 2$ et $b = 1$ donc $b - a = 1 - 2 = \boxed{-1}$

La quantité $b - a$ est décroissante, positive puis devient négative : on sort de la boucle.

Coût, efficacité, complexité d'un algorithme

Introduction

Déterminer l'efficacité d'un algorithme est important. Certaines instructions sons répétées de nombreuses fois et peuvent finir par prendre beaucoup de temps. On cherche alors à calculer un ordre de grandeur du nombre de calculs réalisés.

1. La recherche d'une valeur dans un tableau (minimum, maximum) est de complexité **linéaire**. Le nombre de calculs (comparaisons) est proportionnel à la dimension n du tableau. On note cette complexité par $O(n)$.
2. Le tri d'un tableau par sélection ou insertion est de complexité **quadratique**. Le nombre d'opérations est proportionnel au carré de la dimension n du tableau. On note cette complexité par $O(n^2)$.
3. La recherche par dichotomie est de complexité logarithmique. On la note $O(\log_2(n))$.

Propriété

On peut comparer l'efficacité des algorithmes en comparant leur complexité. Pour un tableau de dimension n, on a :

$$O(\log_2(n)) < O(n) < O(n^2)$$