

Recherche dichotomique



Objectifs

- But des algorithmes de recherche
- Principes et précondition de la recherche dichotomique
- Algorithme
- Évaluation

Algorithmes de recherche

- Le but est **d'obtenir une sous-collection** de valeurs dans **une collection** de valeurs **par rapport à un critère**
- **Une version plus restrictive** que nous développons ici est de rechercher **si une valeur est présente dans une collection**
- Le retour de l'algorithme est généralement de deux natures :
 - **Un entier** qui représente la position du premier élément trouvé, -1 si non trouvé
 - **Un booléen** qui indique si la valeur est présente ou non

Algorithme intuitif V. 1 – Rappels

```
booleen RechercherValeur(int[] p_collection, int p_valeurAChercher) {
    booleen estTrouvee = faux;
    pour entier indiceValeurCourante de 0 à p_collection.Capacite - 1 faire {
        si (p_collection[indiceValeurCourante] == p_valeurAChercher) {
            estTrouvee = vrai;
        }
    }
    renvoyer estTrouvee;
}
```

Algorithme intuitif V. 2 – Rappels

```
booleen RechercherValeurOptimisee(int[] p_collection, int p_valeurAChercher) {
    booleen estTrouvee = faux;
    entier indiceValeurCourante = 0;
    tant que (non estTrouvee et indiceValeurCourante < p_collection.Capacité) faire {
        si (p_collection[indiceValeurCourante] == p_valeurAChercher) {
            estTrouvee = vrai;
        }
        ++indiceValeurCourante;
    }

    renvoyer estTrouvee;
}
```

Principes de la recherche dichotomique

- La recherche dichotomique, ou recherche par dichotomie (en anglais : binary search), est un algorithme de recherche pour trouver la position d'un élément dans un tableau trié.
- Le principe est le suivant : comparer l'élément avec la valeur de la case au milieu du tableau ; si les valeurs sont égales, la tâche est accomplie, sinon on recommence dans la moitié du tableau pertinente.

Principes de la recherche dichotomique

Soit le tableau trié suivant :

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42

Principes de la recherche dichotomique

Recherche de -5

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42

Principes de la recherche dichotomique

Recherche de -5

$3 == -5 \Rightarrow \text{faux}$

$3 < -5 \Rightarrow \text{faux}$

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



$$\text{indiceDernier} = (0 + 10) / 2 \Rightarrow 5$$

 indicePremier

 indiceDernier

 indiceMilieu

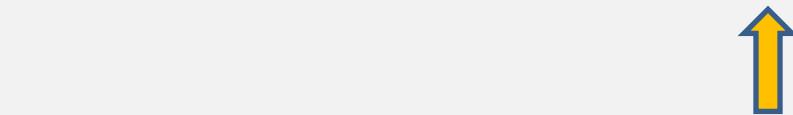
Principes de la recherche dichotomique

Recherche de -5

$3 == -5 \Rightarrow \text{faux}$

$3 < -5 \Rightarrow \text{faux}$

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



indicePremier
↓

indiceDernier
↓

indiceMilieu
↑

Principes de la recherche dichotomique

Recherche de -5

$-13 == -5 \Rightarrow \text{faux}$

$-13 < -5 \Rightarrow \text{vrai}$

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



$$\text{indiceDernier} = (0 + 4) / 2 \Rightarrow 2$$

indicePremier

indiceDernier

indiceMilieu

Principes de la recherche dichotomique

Recherche de -5

$-13 == -5 \Rightarrow \text{faux}$

$-13 < -5 \Rightarrow \text{vrai}$

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



indicePremier

indiceDernier

indiceMilieu

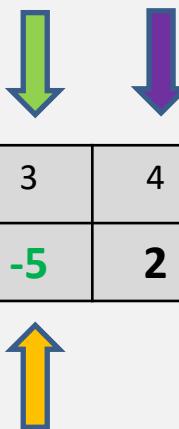
Principes de la recherche dichotomique

Recherche de -5

$-5 == -5 \Rightarrow \text{vrai}$

valeur trouvée !

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



$$\text{indiceDernier} = (3 + 4) / 2 \Rightarrow 3$$

 indicePremier

 indiceDernier

 indiceMilieu

Principes de la recherche dichotomique

Recherche de -3

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42

Principes de la recherche dichotomique

Recherche de -3

$3 == -3 \Rightarrow \text{faux}$

$3 < -3 \Rightarrow \text{faux}$

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



$$\text{indiceDernier} = (0 + 10) / 2 \Rightarrow 5$$

 indicePremier

 indiceDernier

 indiceMilieu

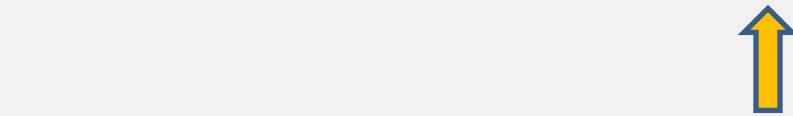
Principes de la recherche dichotomique

Recherche de -3

$3 == -3 \Rightarrow \text{faux}$

$3 < -3 \Rightarrow \text{faux}$

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



 indicePremier

 indiceDernier

 indiceMilieu

Principes de la recherche dichotomique

Recherche de -3

$-13 == -3 \Rightarrow \text{faux}$

$-13 < -3 \Rightarrow \text{vrai}$

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



$$\text{indiceDernier} = (0 + 4) / 2 \Rightarrow 2$$

indicePremier

indiceDernier

indiceMilieu

Principes de la recherche dichotomique

Recherche de -3

$-13 == -3 \Rightarrow \text{faux}$

$-13 < -3 \Rightarrow \text{vrai}$

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



indicePremier

indiceDernier

indiceMilieu

Principes de la recherche dichotomique

Recherche de -3

$-5 == -3 \Rightarrow \text{faux}$

$-5 < -3 \Rightarrow \text{vrai}$

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42

Diagram illustrating the search process:

- A green arrow points down to the value **-5**.
- A purple arrow points down to the index **4**.
- A yellow double-headed arrow indicates the search range between indices **3** and **4**.

$$\text{indiceDernier} = (3 + 4) / 2 \Rightarrow 3$$

 indicePremier  indiceDernier  indiceMilieu

Principes de la recherche dichotomique

Recherche de -3

$-5 == -3 \Rightarrow \text{faux}$

$-5 < -3 \Rightarrow \text{vrai}$

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



indicePremier

indiceDernier

indiceMilieu

Principes de la recherche dichotomique

Recherche de -3

$2 == -3 \Rightarrow \text{faux}$

$2 < -3 \Rightarrow \text{faux}$

The diagram illustrates a binary search algorithm. At the top, two green arrows point downwards towards the center of an array. Below the array, a yellow double-headed arrow points upwards from the center. The array itself consists of two rows of 11 cells each, indexed from 0 to 10 above the first row. The first row contains the indices, and the second row contains the corresponding values. The value at index 4 is highlighted in bold black as 2.

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42

$$\text{indiceDernier} = (4 + 4) / 2 \Rightarrow 4$$

indicePremier

indiceDernier

indiceMilieu

Principes de la recherche dichotomique

Recherche de -3

indicePremier <= indiceDernier => **faux**

STOP => Non trouvée !

0	1	2	3	4	5	6	7	8	9	10
-99	-18	-13	-5	2	3	7	12	24	39	42



indicePremier

indiceDernier

indiceMilieu

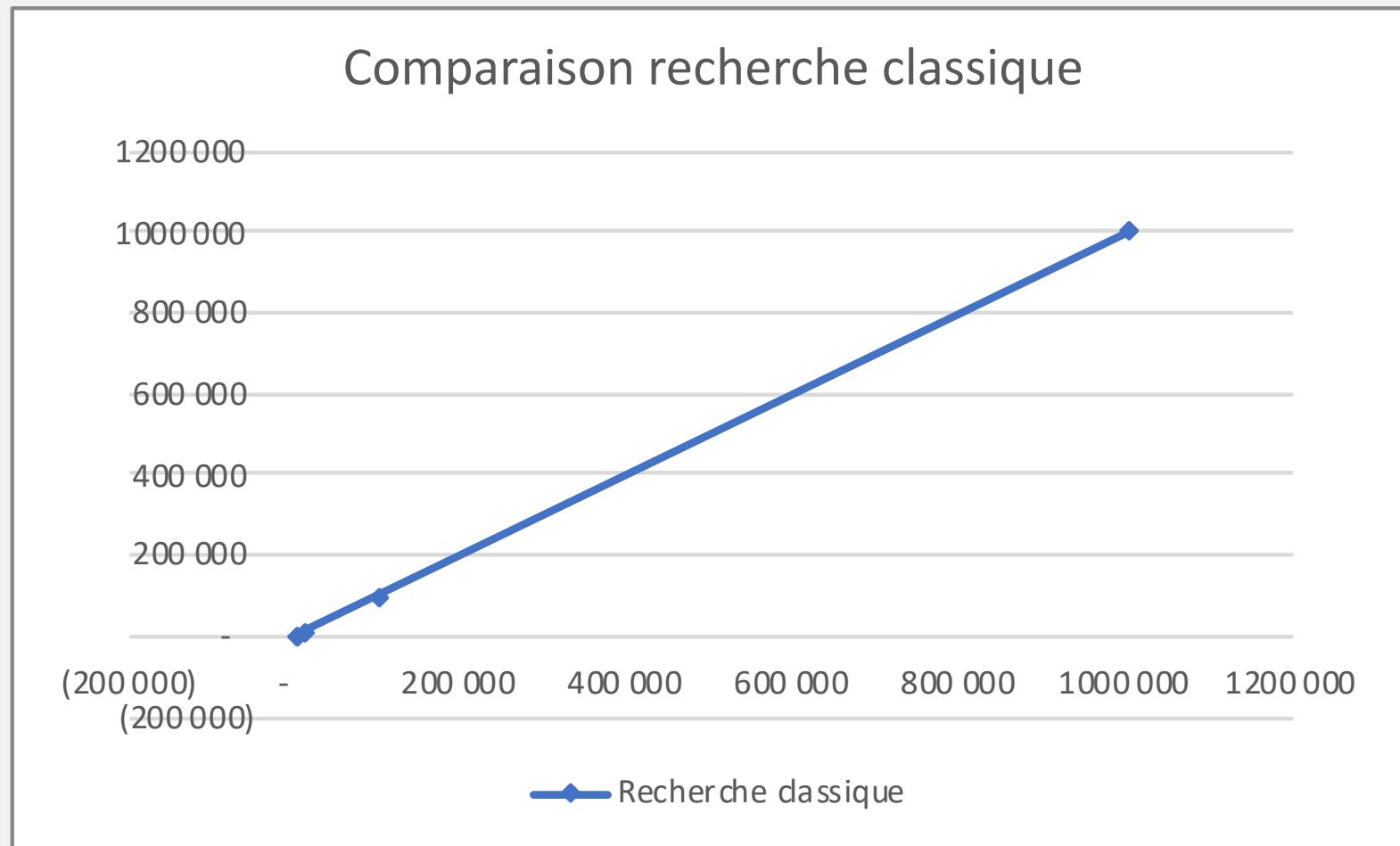
Algorithme de recherche dichotomique

```
boolean RechercherValeurDichotomie(int p_collection, int p_valeurAChercher) {
    boolean estTrouvee = faux;
    entier indicePremier = 0;
    entier indiceDernier = p_collection.Capacité - 1;
    entier indiceMilieu = 0;

    tant que (non estTrouvee et indicePremier <= indiceDernier) {
        indiceMilieu = (indicePremier + indiceDernier) / 2;
        si (p_collection[indiceMilieu] == p_valeurAChercher) {
            estTrouvee = vrai;
        } sinon si (p_collection[indiceMilieu] < p_valeurAChercher) {
            indicePremier = indiceMilieu + 1;
        } sinon {
            indiceDernier = indiceMilieu - 1;
        }
    }

    renvoyer estTrouvee;
}
```

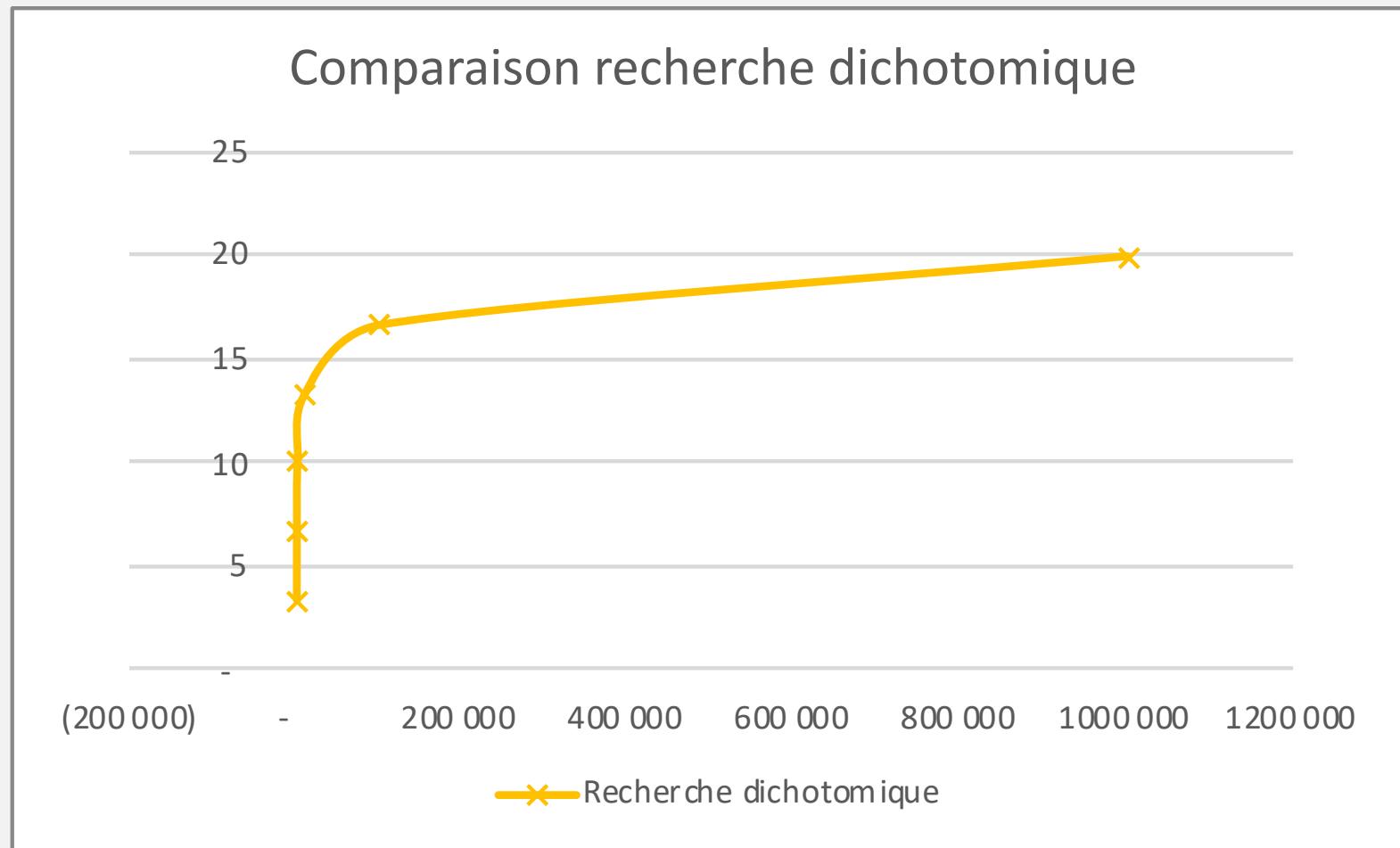
Intuition sur la complexité des algorithmes



x : quantité de données

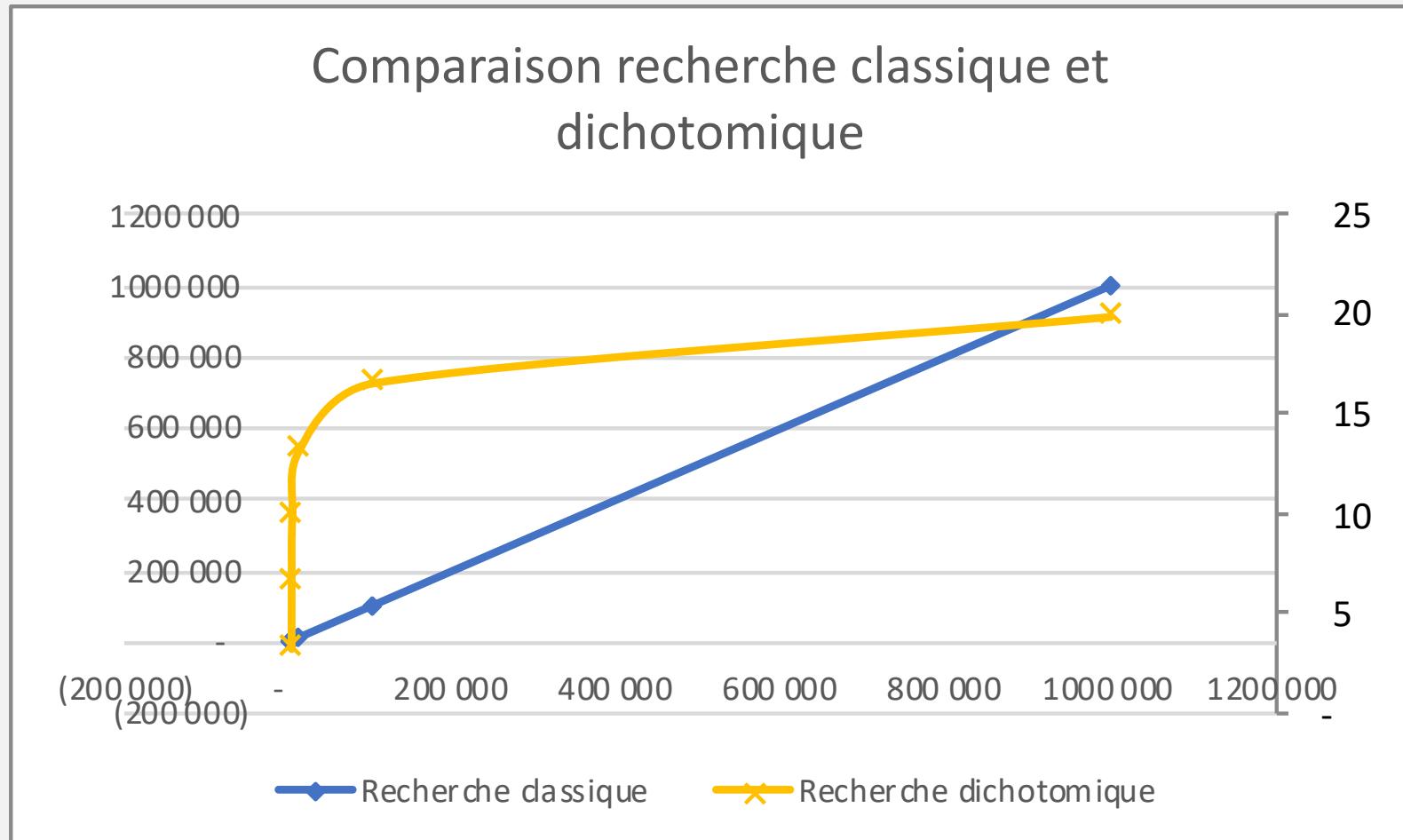
y : nombre de comparaison s

Intuition sur la complexité des algorithmes



x : quantité de données
y : nombre de comparaison s

Intuition sur la complexité des algorithmes



x : quantité de données

y : nombre de comparaison s