

Algorithmique

Partie 2

Première NSI

Lycée Vaugelas, Chambéry

L'algorithme de la recherche dichotomique

Résoudre de plus petites instances du même problème

L'algorithme de la recherche dichotomique

Résoudre de plus petites instances du même problème

Distribuer le fichier `02_algo_eleve.pdf`

→ faire les exercices de la feuille

Les algorithmes de recherche dans un tableau que l'on a vu :

recherche_lineaire	$\mathcal{O}(n)$
meilleure_recherche_lineaire	$\mathcal{O}(n)$
recherche_dichotomique	$\mathcal{O}(\lg n)$

Quelle est la recherche la plus rapide ?

Les algorithmes de recherche dans un tableau que l'on a vu :

recherche_lineaire	$\mathcal{O}(n)$
meilleure_recherche_lineaire	$\mathcal{O}(n)$
recherche_dichotomique	$\mathcal{O}(\lg n)$

Quelle est la recherche la plus rapide ? La **recherche dichotomique**

Pourquoi ?

Les algorithmes de recherche dans un tableau que l'on a vu :

recherche_lineaire	$\mathcal{O}(n)$
meilleure_recherche_lineaire	$\mathcal{O}(n)$
recherche_dichotomique	$\mathcal{O}(\lg n)$

Quelle est la recherche la plus rapide ? La **recherche dichotomique**

Pourquoi ? Parce que le tableau était **trié** !

Si on doit faire des recherches successives dans un tableau très grand, il peut être intéressant de le trier d'abord.

Ce sera ce qu'on fera prochainement : trier des tableaux.
Il existe toute une série d'algorithmes pour trier des tableaux : on en a 2 au programme.

- ▶ tri par insertion
- ▶ tri par sélection

... mais ce sera pour plus tard ! ou pas !

L'algorithme de la recherche dichotomique

Résoudre de plus petites instances du même problème

- ▶ on est bien confronté au même problème (rechercher une valeur dans un sous-tableau)
- ▶ le tableau dans lequel se fait la recherche est de plus en plus petit

→ on est bien en train de résoudre de plus petites instances du même problème

il y a un moyen très pratique de le faire : la **réursion** !

Si on reprend ce qu'on a vu :

`recherche_dichotomique(A,5,12)` sur le tableau
 $A = [-2, 4, 4, 10, 12]$

1ère instance du problème :

sous-tableau de $A = [-2, 4, 4, 10, 12]$ ($q = 3$)

2ème instance du problème :

sous-tableau de $A = [-2, 4, 4, 10, 12]$ ($q = 4$)

3ème instance du problème :

sous-tableau de $A = [-2, 4, 4, 10, 12]$

Algo : `recherche_dichotomique_recursive (A, p, r, x)`

Entrées :

- A : tableau trié
- p, r : on examine le sous-tableau $A[p,r]$
- x : valeur recherchée dans A

Sorties :

- soit indice i tel que $A[i] = x$
- soit valeur spéciale « Non Trouvé »

Procédure :

1. Si $p > r$, alors retourner *Non-Trouvé*
2. Sinon ($p \leq r$), faire la chose suivante :

A. Fixer q à `partie_entiere((p+r)/2)`

B. Si $A[q] = x$, alors retourner q

C. Sinon, si $A[q] > x$, alors retourner

`recherche_dichotomique_recursive (A, p, q-1, x)`

D. Sinon, retourner `recherche_dichotomique_recursive (A, p+1, q, x)`

Les conditions de la récursion

- 1) il faut au moins **un cas de base**
- 2) chaque appel récursif doit se faire sur une **instance plus petite du même problème**