
Algorithmique & programmation

Chapitre 2 : Vecteurs

Vecteur trié

Recherche dichotomique

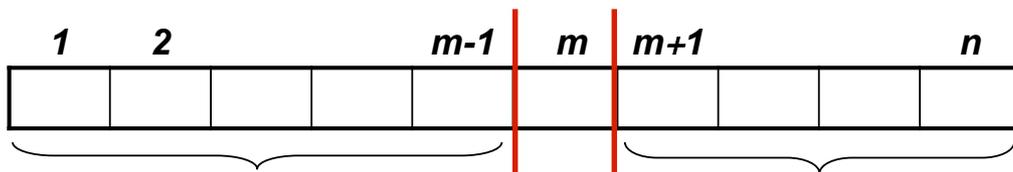
Recherche dichotomique

Recherche de elem dans $V[1..n]$ trié

Idée

■ découper V en trois sous-vecteurs triés tels que

$V[1..m-1] \leq V[m] \leq V[m+1..n]$



■ si $\text{elem} = V[m]$: on a fini

■ si $\text{elem} < V[m]$: question $\text{elem} \in V[1..m-1]$?

■ si $\text{elem} > V[m]$: question $\text{elem} \in V[m+1..n]$?

✓ **Le nouveau problème est plus petit**

Recherche dichotomique

- La taille des vecteurs examinés va diminuer pas à pas vers 0 en un nombre fini de pas
 - soit on trouvera m tel que $V[m] = \text{elem}$
 - soit on trouvera un vecteur vide et $\text{elem} \notin V$

- Comment choisir m pour découper $V[1..n]$ ainsi que les autres sous-vecteurs ?
 - si $V[\text{inf}..\text{sup}]$ est le vecteur à découper, on choisit :
 - $m := (\text{inf} + \text{sup}) \text{ div } 2$ (div : division entière)
 - on divise la taille du vecteur (du problème) par 2

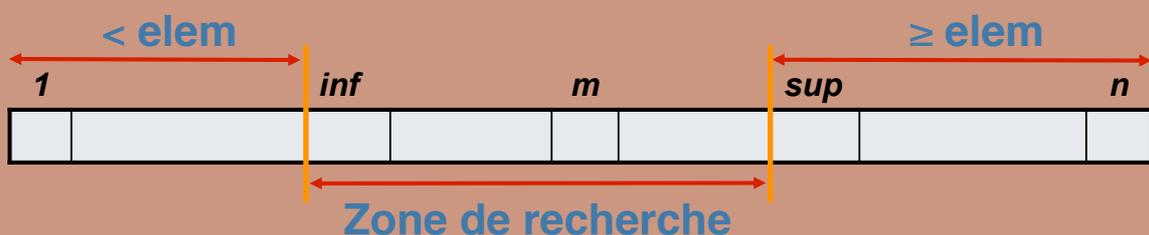
Recherche dichotomique

- Plusieurs hypothèses possibles ...

- $V[1..\text{inf}-1] < \text{elem} < V[\text{sup}+1..n]$



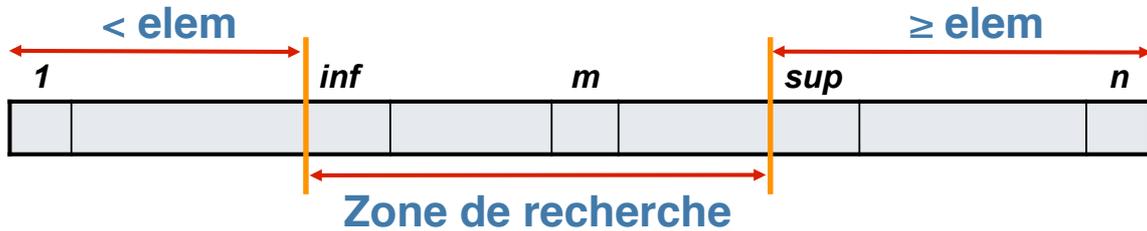
- $V[1..\text{inf}-1] < \text{elem} \leq V[\text{sup}..n]$



Recherche dichotomique (version 2)

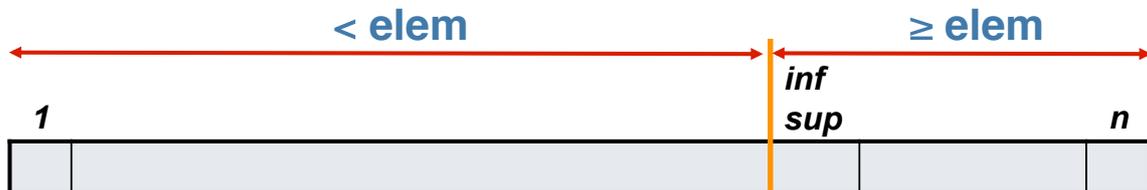
□ Travailler à partir de la situation suivante :

■ $V[1..inf-1] < elem \leq V[sup..n]$



□ À la fin (i.e. la zone de recherche est vide)

■ on est dans la situation suivante



Recherche dichotomique (version 2)

□ Situation final (rappel)



□ Il faudra donc vérifier que $elem = V[sup]$ pour savoir si l'élément est présent ou non

fonction `dicho` ($d V[1..n]$: vecteur ; $d elem$: t) : entier ;

spécification $\{V \text{ trié}, n > 0\} \rightarrow \{V[1..m-1] < elem \leq V[m..n], m \in [1..n],$
(résultat = m, $V[m] = elem$) \vee (résultat = 0, $elem \notin V\}$

Recherche dichotomique (version 2)

□ Raisonnement par récurrence

Hypothèse $V[1..inf-1] < elem \leq V[sup..n]$

➤ $inf = sup \Leftrightarrow elem \in V$ ssi $V[sup]=elem \Leftrightarrow$

➤➤ $V[sup] = elem \Leftrightarrow \text{résultat} = sup ; *$

➤➤ $V[sup] \neq elem \Leftrightarrow \text{résultat} = 0 ; *$

➤ $inf < sup \Leftrightarrow$

$m := (inf + sup) \text{ div } 2 ;$

➤➤ $V[m] \geq elem \Leftrightarrow sup := m ; \Rightarrow H$

➤➤ $V[m] < elem \Leftrightarrow inf := m + 1 ; \Rightarrow H$

Trouver l'itération ?

Chemin depuis le retour à l'hypothèse vers les conditions

Recherche dichotomique (version 2)

□ Raisonnement par récurrence

Hypothèse $V[1..inf-1] < elem \leq V[sup..n]$

➤ $inf = sup \Leftrightarrow elem \in V$ ssi $V[sup]=elem \Leftrightarrow$

➤➤ $V[sup] = elem \Leftrightarrow \text{résultat} = sup ; *$

➤➤ $V[sup] \neq elem \Leftrightarrow \text{résultat} = 0 ; *$

➤ $inf < sup \Leftrightarrow$

$m := (inf + sup) \text{ div } 2 ;$

➤➤ $V[m] \geq elem \Leftrightarrow sup := m ; \Rightarrow H$

➤➤ $V[m] < elem \Leftrightarrow inf := m + 1 ; \Rightarrow H$

Itération tantque $(inf < sup)$ faire ...

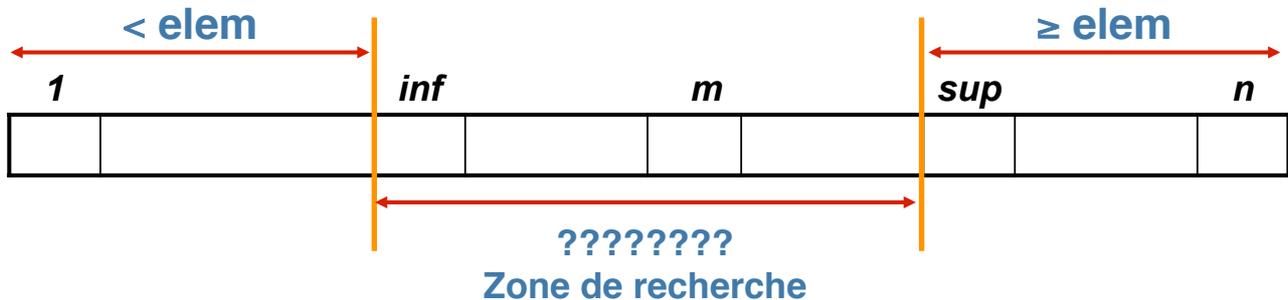
Trouver l'initialisation ?

Dans quel sous-vecteur doit-on commencer à chercher ?

Recherche dichotomique (version 2)

□ Trouver l'initialisation

■ Rappel



■ Si $elem > V[n]$

- il ne faut pas commencer la recherche

■ Sinon

- la « zone de recherche » initiale est tout le vecteur
- donc $inf := 1$ et $sup := n$

Recherche dichotomique (version 2)

□ Raisonnement par récurrence

Hypothèse $V[1..inf-1] < elem \leq V[sup..n]$

➤ $inf = sup \Leftrightarrow elem \in V \text{ ssi } V[sup] = elem \Leftrightarrow$

➤➤ $V[sup] = elem \Leftrightarrow \text{résultat} = sup ; *$

➤➤ $V[sup] \neq elem \Leftrightarrow \text{résultat} = 0 ; *$

➤ $inf < sup \Leftrightarrow$

$m := (inf + sup) \text{ div } 2 ;$

➤➤ $V[m] \geq elem \Leftrightarrow sup := m ; \blacktriangleright H$

➤➤ $V[m] < elem \Leftrightarrow inf := m + 1 ; \blacktriangleright H$

Itération tantque $(inf < sup)$ faire ...

Initialisation ➤ $elem > V[n] \Leftrightarrow elem \notin V \Leftrightarrow \text{résultat} = 0 ; *$

➤ $elem \leq V[n] \Leftrightarrow inf := 1 ; sup := n ; \blacktriangleright H$

Fonction dichotomie

```
fonction dichotomie (d V[1..n] : vecteur ; d elem : t) : entier ;
    m, inf, sup : entier ;
debfonction
    si elem > V[n] alors
        retour 0 ;
    sinon
        inf := 1 ; sup := n ;
        tantque inf < sup faire
            m := (inf + sup) div 2 ;
            si elem ≤ V[m] alors
                sup := m ;
            sinon
                inf := m+1 ;
            finsi ;
        finfaire ;
        si elem = V[sup] alors
            retour sup ;
        sinon
            retour 0 ;
        finsi ;
    finfonction ;
```

$\{V[1..inf-1] < elem [V[sup ..n] \}$

$\{elem [V[sup ..n] \}$

$\{V[1..inf-1] > elem \}$

$\{inf = sup, V[1..inf-1] < elem \leq V[sup ..n] \rightarrow V[1..sup-1] < elem \leq V[sup ..n] \}$

Fonction dichotomie

```
function dichotomie (V : in vectInt ; elem : in integer) return integer is
    m, inf, sup : integer ;
begin
    if elem > V(V'Last) then
        retour 0 ;
    else
        inf := V'First ; sup := V'Last ;
        while inf < sup loop
            m := (inf + sup) / 2 ;
            if elem ≤ V(m) then
                sup := m ;
            else
                inf := m+1 ;
            end if ;
        end loop ;
        if elem = V(sup) then
            return sup ;
        else
            return 0 ;
        end if ;
    end if ;
end dichotomie ;
```

n de V[1..n] {borne supérieure}

1 de V[1..n] {borne inférieure}