

Recherche dichotomique dans un tableau trié

algorithme de recherche dans un tableau trié

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

```
int deb, fin
```

```
// si  $e$  est dans le tableau alors son indice est dans  
[deb, fin]
```

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

```
int deb, fin
```

```
// si  $e$  est dans le tableau alors son indice est dans  
[deb, fin]
```

```
deb  $\leftarrow$  1; fin  $\leftarrow$   $n$ ;
```

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

int deb, fin

// si e est dans le tableau alors son indice est dans
 $[deb, fin]$

$deb \leftarrow 1; fin \leftarrow n;$

tant que $deb \leq fin$

faire

|

fin

retourner

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

int deb, fin

// si e est dans le tableau alors son indice est dans
 $[deb, fin]$

$deb \leftarrow 1$; $fin \leftarrow n$;

tant que $deb \leq fin$

faire

$mil \leftarrow (deb + fin) \text{ div } 2$

si $T[mil] = e$ **alors retourner** *vrai* ;

fin

retourner

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

int deb, fin

// si e est dans le tableau alors son indice est dans
[deb, fin]

$deb \leftarrow 1$; $fin \leftarrow n$;

tant que $deb \leq fin$

faire

$mil \leftarrow (deb + fin) \text{ div } 2$

si $T[mil] = e$ **alors retourner vrai** ;

si **alors**

 |

sinon

 |

fin

fin

retourner

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

int deb, fin

// si e est dans le tableau alors son indice est dans
[deb, fin]

$deb \leftarrow 1$; $fin \leftarrow n$;

tant que $deb \leq fin$

faire

$mil \leftarrow (deb + fin) \text{ div } 2$

si $T[mil] = e$ **alors retourner vrai** ;

si **alors**

 |

sinon

 |

fin

fin

retourner

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

int deb, fin

// si e est dans le tableau alors son indice est dans
[deb, fin]

$deb \leftarrow 1$; $fin \leftarrow n$;

tant que $deb \leq fin$

faire

$mil \leftarrow (deb + fin) \text{ div } 2$

si $T[mil] = e$ **alors retourner** *vrai* ;

si $T[mil] < e$ **alors**

 |

sinon

 |

fin

fin

retourner

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

int deb, fin

// si e est dans le tableau alors son indice est dans
[deb, fin]

$deb \leftarrow 1$; $fin \leftarrow n$;

tant que $deb \leq fin$

faire

$mil \leftarrow (deb + fin) \text{ div } 2$

si $T[mil] = e$ **alors retourner** vrai ;

si $T[mil] < e$ **alors**

$deb \leftarrow 1 + mil$

sinon

fin

fin

retourner

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

```
int deb, fin
```

```
// si  $e$  est dans le tableau alors son indice est dans  
[deb, fin]
```

```
deb  $\leftarrow 1$ ; fin  $\leftarrow n$ ;
```

```
tant que deb  $\leq$  fin
```

```
faire
```

```
    mil  $\leftarrow (deb + fin) \text{ div } 2$ 
```

```
    si  $T[mil] = e$  alors retourner vrai ;
```

```
    si  $T[mil] < e$  alors
```

```
        | deb  $\leftarrow 1 + mil$ 
```

```
    sinon
```

```
        | fin  $\leftarrow mil - 1$ 
```

```
    fin
```

```
fin
```

```
retourner
```

algorithme de recherche dans un tableau trié

Données: un entier e , un tableau trié T de n cases

Résultat: un booléen indiquant si $e \in T$

```
int deb, fin
```

```
// si  $e$  est dans le tableau alors son indice est dans  
[deb, fin]
```

```
deb  $\leftarrow 1$ ; fin  $\leftarrow n$ ;
```

```
tant que deb  $\leq$  fin
```

```
faire
```

```
    mil  $\leftarrow (deb + fin) \text{ div } 2$ 
```

```
    si  $T[mil] = e$  alors retourner vrai ;
```

```
    si  $T[mil] < e$  alors
```

```
        | deb  $\leftarrow 1 + mil$ 
```

```
    sinon
```

```
        | fin  $\leftarrow mil - 1$ 
```

```
    fin
```

```
fin
```

```
retourner faux
```