

Traitements de recherche dans un tableau à une dimension

Cadre :

Comme nous trions les éléments du tableau
(plutôt que de manipuler leurs adresses,),
ce qui suit n'est pas optimisé,
et sert uniquement à des fins pédagogiques :

C'est une première approche.

Plusieurs méthodes sont possibles, et la meilleure méthode
doit être judicieusement choisie dans chaque cas

Le nombre d'opération est essentiel, car le temps minimum
est le critère le plus visible.

Tableau non trié à une seule dimension : au moins une fois ?

Méthode : on compare les éléments du tableau 1 à 1
jusqu'à ce qu'on trouve la valeur recherchée,
ou jusqu'à ce qu'on arrive au dernier élément rempli du tableau.

Programme au moins une fois dans tableau à une dimension

Variables Tableau de nombre_eleves entiers: notes[]
Entier : L, note_cherchee, max

Début

Répéter

Écrire (nombre d'élèves ?) [valeur maxi : max]

Lire : nombre_eleves

Tant que (nombre_eleves<0) ou (nombre_eleves>max)

Pour L de 0 à nombre_eleves-1

Écrire (entrer note L+1ème élève)

Lire(note[L])

Fin Pour

Ecrire(quelle note voulez-vous rechercher ?)

Lire(note_cherchee)

(Méthode 1)

```
L<--0
Tant que (note_cherchee != note[L] et L < nombre_eleves)
    L <-- L+1
Fin Tant que

Si note_cherchee = note[L]
    alors écrire ( il y a au moins 1 étudiant qui a eu note_cherchee/20 )
    sinon écrire (personne n'a eu cette note)
Fin si
```

(Méthode2)

```
L <-- -1

Répéter
    L <-- L+1

Tant que note_cherchee != notes[L]
    et L < nombre_eleves

Si note_cherchee = note[L]
    alors écrire (il y a au moins 1 qui a .. )
    sinon écrire (il n'y a pas...)

Fin Si
```

Dans un ensemble trié à une dimension : combien de fois ?

On déclare une variable de type entier,
qui contiendra le nombre d'occurrences du critère recherché.

Tout le tableau devra être parcouru pour que cette méthode soit valide.

Exemple : nombre_fois

```
nombre_fois <-- 0
```

```
Écrire (quelle note ?)
```

```
Lire( note_cherchee)
```

```
Pour i de 0 à nombre_eleves-1
```

```
    Si note[i] = note_cherchee
```

```
        Alors incrémenter nombre_fois
```

```
        Sinon rien
```

```
    Fin si
```

```
Fin pour
```

```
Écrire (il y a nombre_fois occurrences)
```

Dans un ensemble trié à une seule dimension : au moins une fois ?

Exercice :

On va comparer les éléments du tableau à la valeur recherchée, jusqu'à ce qu'on trouve dans le tableau un élément supérieur ou égal à celle-ci, ou qu'on arrive au dernier élément.

Cas possible à envisager : la note demandée est trop grande, et n'est pas dans le tableau.

Algorithme :

Ecrire (note ?)

Lire (note_cherchée)

L <-- -1

Variable : booléenne trouve <-- faux

Répéter

Si note_cherchée != note[L]

alors incrémenter L

sinon trouve <-- vrai

Fin Si

Tant que (L < nombre_eleves et trouve == faux)

Si trouve = vrai

alors Écrire (un étudiant au moins a eu cette note)

sinon Écrire (pas trouvé)

Fin Si

Dans un ensemble trié à une dimension : recherche dichotomique (traité en TP et en TD)

Attention : cette méthode n'est utilisable
que pour un ensemble déjà trié sur le critère de recherche

Méthode :

On compare la valeur recherchée qui se trouve au milieu de la valeur de l'ensemble

On décide que la valeur recherchée est, de façon exclusive :

- soit dans la première partie de cet ensemble
- soit dans la seconde.

Une fois la moitié contenant la variable déterminée,
on ne conserve que la moitié pouvant contenir la valeur.

Et on réitère la recherche

Tant que le critère d'arrêt fixé au début n'est pas atteint, ou tant que l'ensemble de recherche ne se réduit pas à un seul élément, on réitère la recherche.

Exemple :

Soit le tableau de notes suivant :

1	2	4	6	6	9	12	12	15	16	17	19
---	---	---	---	---	---	----	----	----	----	----	----

1 2 3 4 5 6 7 8 9 10 11 12

Variables

tableau de 12 entiers Tab[12]

entier min = 1

entier max=12

entier note_cherchee

entier i

Début

Écrire (note recherchée ?)

Lire note_cherchee

i <-- (max + min) / 2

tant que T[i] != note_cherchee ou i != min

si T[i] < note_cherchee

alors min <-- i

sinon max <-- i

fin si

 i <-- (max+min)/2

fin tant que

si T[i] = note_cherchee

alors écrire (note_cherchee est dans l'ensemble)

sinon écrire (pas trouvé)

Fin si

Fin