

# Les tableaux ( suite)

Recherche d'un élément dans un tableau

Tableaux à deux dimensions

# Recherche d'un élément dans un tableau

- **Recherche d'un élément dans un tableau**

- On recherche la première occurrence d'une valeur val.
- Cette valeur n'est pas forcément présente dans le tableau.
- Soit le tableau : **tab**[**taille**] d'entiers par exemple.

Les algorithmes suivants sont également utilisables sur des tableaux de réels, de caractères.

## **Recherche séquentielle avec un tableau non trié : parcours intégral**

- on parcourt le tableau case par case ;
- on s'arrête quand on a parcouru tout le tableau ou quand on trouve la valeur ;
- on affiche le numéro de la case de la valeur si elle est trouvée.

# Programme Java de recherche séquentielle

```
Public class tbNtrie{

    Public static void recherche(){
        int i=0 ;
        int val=10;

        while(i<taille && tab[i] !=val){ i++ ;
        }

        if(i<taille)
            System.out.println(« valeur trouvée »+ val + « case : »+i) ;
        else
            System.out.println("valeur non trouvée") ;
    }
}
```

# Recherche séquentielle dans un tableau trié

## Recherche séquentielle dans un tableau trié :

- on parcourt le tableau case par case ;
- on s'arrête quand on a parcouru tout le tableau ou quand on trouve la valeur, ou quand on est sur une valeur supérieure à la valeur cherchée (avantage du tableau trié) ;
- on affiche le numéro de la case de la valeur si elle est trouvée.

```
public class tbTrie{  
  
    public static void recherche(){  
        int i=0 ;  
        while(i<taille && tab[i] <val){ i++ ;  
        }  
  
        if(tab[i] ==val)  
            System.out.println("valeur trouvée") ;  
        else  
            System.out.println("valeur non trouvée") ;  
    }  
}
```

# Recherche par dichotomie

## Recherche par dichotomie (tableau trié)

On ne parcourt plus le tableau case par case, mais moitié par moitié ;

Principe : on coupe le tableau en deux parties égales. On calcule  $m$  l'indice de la moitié du tableau ( $m = (\text{inf} + \text{sup}) / 2$ ) et on cherche la valeur dans la partie concernée selon les bornes inf, sup et le milieu ; puis on redivise la partie concernée en deux sous-parties jusqu'à obtenir la valeur ;

on s'arrête quand on trouve la valeur, ou quand les bornes inf et sup sont inversées ( $\text{inf} > \text{sup}$ ) ;

on affiche le numéro de la case de la valeur si elle est trouvée

# Recherche par dichotomie

```
Public class tbTrie{

    Public static void recherche(){
    int val;
    int inf=0 ;int sup=taille-1 ; int m=(sup+inf)/2
        while(inf<sup && tab[m]!=val){
            if(val<tab[m]) sup=m-1 ;
            else {
                inf=m+1 ;}
            m=(sup+inf)/2;
        }

        if(inf>sup)
            System.out.println("valeur non trouvée");
        else
            System.out.println(val + " case " +m) ;
    }
}
```

# Tableaux à deux dimensions

## Contexte :

- On souhaite créer un fichier de notes de 32 élèves de BTS SIO dans 7 matières différentes en vue de calculer la moyenne de chaque étudiant.
- Doit-on créer un tableau de 7 cases pour chaque élève ? On aurait alors 32 tableau à gérer !

La solution serait de créer un tableau de 32 lignes et 8 colonnes de ce type :

	<b>Module 1</b>	<b>Module 2</b>	<b>.....</b>	<b>Module 8</b>
Etud1				
Etud2				
....	....	....	....	...
Etud32				

Nous avons donc opté pour un tableau à deux dimensions. Chaque case du tableau est repérée par deux indices : son numéro de ligne et son numéro de colonne. Le premier indice est toujours 0. La première case a donc pour adresse [0][0]

# Déclaration d'un tableau à deux dimensions

- En reprenant l'exemple précédent, pour déclarer le tableau de notes des étudiants, on procède comme suit :

`int [][] tabNotes=new int [32][8];` ou bien `int tabNotes [][] =new int [32][8];` pour un tableau d'entiers. On peut remplacer `int` par `float`, `double`, `char` ou `String`.

Les étudiants sont identifiés par un numéro. Comment peut-on stocker leur nom ? Ne pas oublier que les cases doivent toutes contenir des valeurs de même type.

On ne *peut* donc *pas* rajouter une colonne où l'on aurait le nom de chaque étudiant (car on aurait un mélange de chaînes et de réels).

La seule solution consiste à définir un tableau à une dimension, de longueur *NbEtud*, dont la case *i* contient le nom de l'étudiant *i*. Pour stocker des données de types différents, il faut des tableaux différents (même si ces données correspondent sémantiquement à la même entité).



# Exemples

- 1) Ecrire un programme qui saisit un nombre entre 1 et 12 et indique en toutes lettres à quel mois cela correspond. Vous devez conserver les mois dans un tableau.
- 2) Ecrire un programme qui permet de générer un tableau de 5 lignes et 5 colonnes qui contient des entiers aléatoires compris entre 1 et 25 et qui permet de calculer le nombre d'occurrences du nombre 10.

- Considérons l'exemple suivant : soit une section de 1000 étudiants qui suivent 7 modules, on veut calculer la moyenne de chaque étudiant. Quelles sont les données ? Quelles sont les résultats ? comment on les déclare ?
  - ❑ Utiliser  $1000 * 7$  variables pour les notes de 1000 étudiants dans 7 modules et 1000 variables pour les moyennes !!!!! c'est déraison,
  - ❑ Utiliser 7 vecteurs (tableau à une dimension) de taille 1000 pour les notes et un autre pour les moyennes c'est aussi illogique.
  - ❑ Dans ce cas le vecteur ne répond pas à cette réalité, alors chercher d'autres structures
  - ❑ Tableau à deux dimension (matrice), nous créons une seule variable de type matrice de 1000 lignes (étudiants) et 8 colonnes (notes + moyenne).
  - ❑ Une table ou matrice est un tableau bidimensionnel (à deux dimension), composé de plusieurs lignes et plusieurs colonnes.
  - ❑ Pour se déplacer dans une matrice, on a besoin de deux indices (par convention i et j), le premier indique le numéro de la ligne et le deuxième indique le numéro de la colonne. L'élément d'indice [i,j] est celui du croisement de la ligne i avec la colonne j.
  - ❑ Une case dans une matrice est connue par le nom de la matrice suivie entre crochets par son indice de ligne, son indice de colonne : Exemple M[i,j]
  - ❑ Comme les vecteurs :
  - ❑ L'indice peut être une constante, une variable ou une expression arithmétique.
  - ❑ Un élément de la matrice est manipulé exactement comme une variable.
  - ❑ Dans la partie Constante, on peut définir la taille de la matrice (lignes, colonnes). Ensuite, on peut déclarer le nombre d'éléments réellement à saisir.
  - ❑ Le nombre d'éléments à saisir ne doit pas dépasser la taille de la matrice.
  - ❑ En Algorithmique l'indice des lignes et des colonnes commence par 1.