



Programmation C - Les opérations sur les tableaux

Christel DARTIGUES-PALLEZ

Quoi faire avec des tableaux?

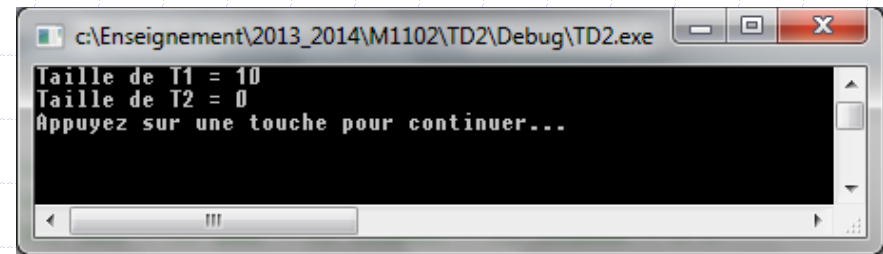
- ◆ Rechercher des propriétés
 - Tableau vide? plein?
 - Tableau trié?
 - Valeur min? max? moyenne?
- ◆ Rechercher une valeur
- ◆ Trier
- ◆ Supprimer une valeur
- ◆ Insérer une valeur

Rechercher des propriétés

◆ Tableau vide

- Taille déclarée d'un tableau \neq nombre d'éléments effectivement insérés dans le tableau
- Toujours garder une valeur qui donne la taille réelle d'un tableau
- Exemple

```
void exempleTailleTableau() {  
  
    int i, T1[100], T2[10], n1=10, n2= 0 ;  
    for (i=0; i<n1; i++)  
        T1[i] = i ;  
    printf ("Taille de T1 = %d\n", n1) ;  
    printf ("Taille de T2 = %d\n", n2) ;  
}
```



Rechercher des propriétés

◆ Tableau trié, min, max, moyenne

- Faut-il absolument parcourir tout le tableau?
- Non
 - ◆ Pour ce qui est de savoir si le tableau est trié
 - ◆ Dès qu'on a une valeur qui indique que le tableau n'est pas trié, on DOIT s'arrêter
- Oui
 - ◆ Pour calculer le min, le max et la moyenne
 - ◆ Le min ou le max peuvent se trouver n'importe où, y compris à la dernière case du tableau

Rechercher une valeur

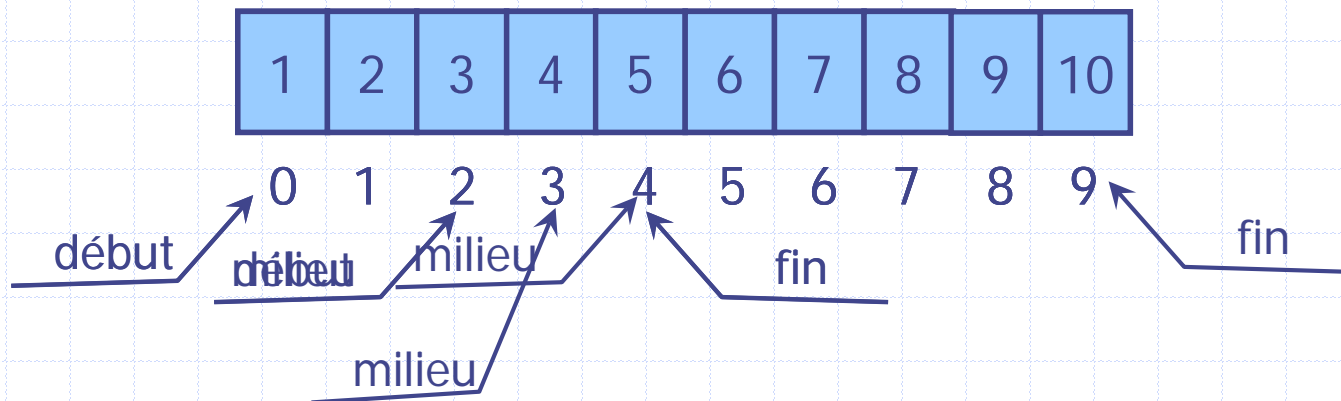
- ◆ On ne DOIT pas parcourir tout le tableau
- ◆ Algo de recherche classique

```
int exempleRechercheValeur (int T [], int nb, int v) {  
    while (valeur non trouvée) && (pas à la fin du tableau)  
        on regarde la case suivante du tableau  
  
    if (je suis sortie parce que la première condition est fausse)  
        return (1) ; // la valeur est présente  
    else  
        return (0) ; // la valeur n'est pas présente  
}
```

- ◆ Que se passe-t-il dans le pire des cas?
 - Tableau d'un million de valeurs, la valeur cherchée est la dernière... 1 million de test

Rechercher une valeur

- ◆ Un algo un peu plus évolué: recherche par dichotomie
 - Hypothèse importante: le tableau DOIT être trié
 - ◆ Exemple: on recherche la valeur 4



Recherche d'une valeur - dichotomie

◆ Que se passe-t-il dans le pire des cas?

- Tableau d'un million de valeurs, la valeur cherchée est la dernière...
- Recherche d'une valeur parmi :
- Au pire : 21 tests !

1 000 000	976
500 000	488
250 000	244
125 000	122
62 500	61
31 250	30
15 625	15
7 812	8
3 906	4
1 953	2

Trier un tableau

- ◆ Il existe beaucoup d'algos de tris, plus ou moins efficaces en terme de complexité
 - Nombre d'opérations effectuées (= temps de calcul)

Tri à bulle

◆ Principe

- On parcourt le tableau et on compare 2 éléments successifs
- Si 2 éléments successifs ne sont pas dans l'ordre croissant, on les échange
- Lorsqu'on a parcouru tout le tableau sans faire d'échange, le tableau est trié et le programme s'arrête

Tri à bulle

11 3 5 1 22 13 8 9

3 11 5 1 22 13 8 9

3 5 11 1 22 13 8 9

3 5 1 11 22 13 8 9

3 5 1 11 13 22 8 9

3 5 1 11 13 8 22 9

3 5 1 11 13 8 9 22

3 1 5 11 13 8 9 22

3 1 5 11 8 13 9 22

3 1 5 11 8 9 13 22

1 3 5 11 8 9 13 22

1 3 5 8 11 9 13 22

1 3 5 8 9 11 13 22

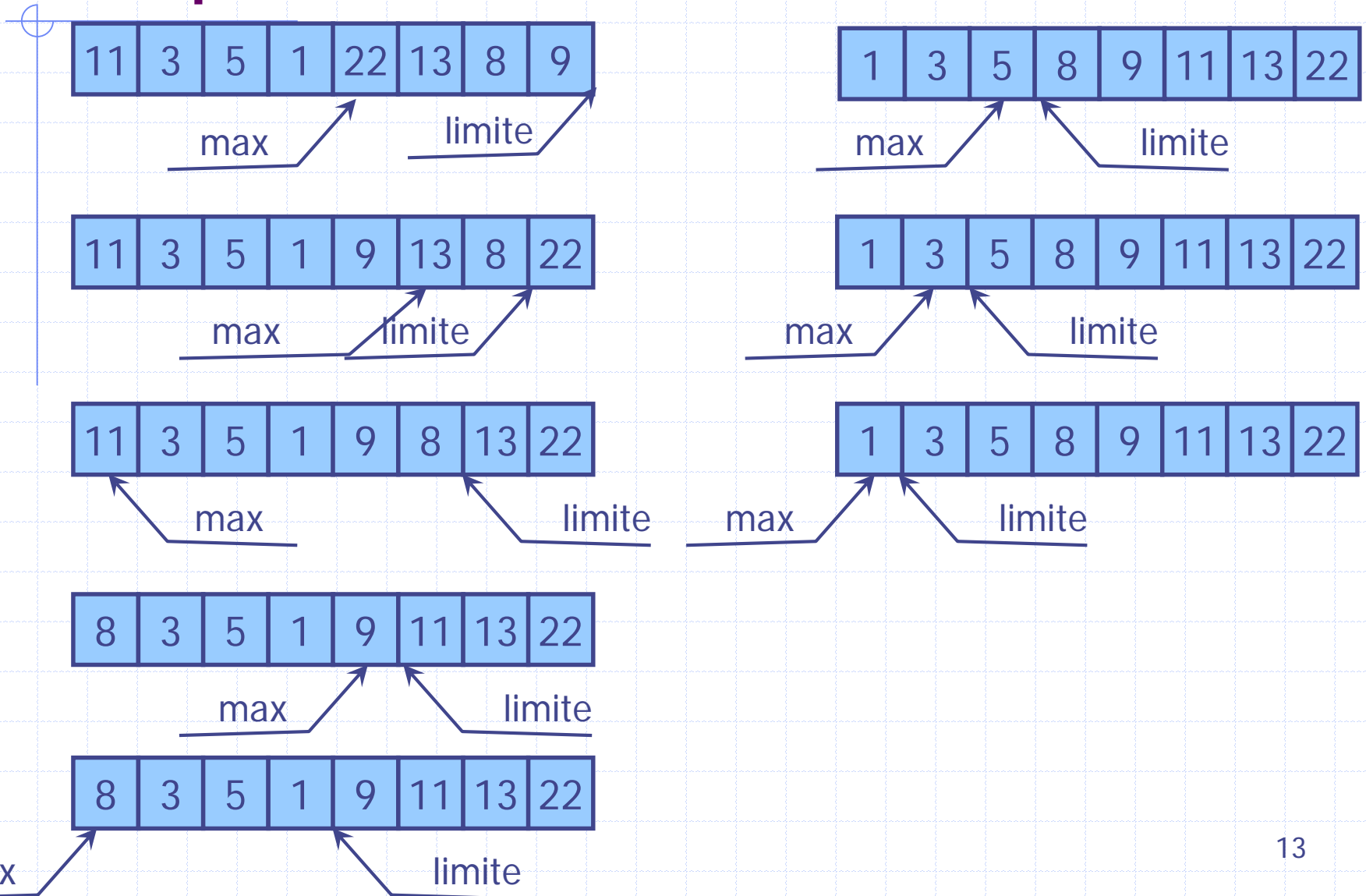
Tri à bulle

```
void triBulle (int T [], int n) {  
    int i = 0;           // pour parcourir mon tableau  
    int trie = 0 ;       // pour arrêter le tri  
    int aux = 0 ;        // pour échanger les valeurs  
  
    system ("CLS") ;  
    afficheTab (T, n) ;  
    printf ("\n\n") ;  
    system ("PAUSE") ;  
  
    while (!trie) {  
        trie = 1 ;  
        for (i=0; i<n-1; i++) {  
            if (T[i]>T[i+1]) {  
                aux = T[i] ;  
                T[i] = T[i+1] ;  
                T[i+1] = aux ;  
                trie = 0 ;  
  
                system ("CLS") ;  
                afficheTab (T, n) ;  
                printf ("\n\n") ;  
                system ("PAUSE") ;  
            }  
        }  
    }  
}
```

Tri par sélection

- ◆ On recherche le plus grand élément du tableau et on le place à la fin (ou le plus petit)
- ◆ On répète cette opération pour les N valeurs du tableau
- ◆ On utilise une limite qui part de la fin du tableau et qui s'arrête au début
 - Cette limite sépare la partie triée du tableau de la partie non encore triée
 - Tous les éléments à droite de la limite sont triés et sont plus grands que les valeurs situées à gauche de cette limite

Tri par sélection



Tri par sélection

```
void triSelection (int T [], int n) {  
    int i = 0 ;           // pour parcourir mon tableau  
    int trie = 0 ;        // pour arrêter le tri  
    int aux = 0 ;         // pour échanger les valeurs  
    int limite = n-1 ;    // pour séparer la partie triée  
    int max = 0 ;         // pour trouver le max de la partie non triée  
  
    system ("CLS") ; afficheTab (T, n) ;  
    printf ("\n\n") ; system ("PAUSE") ;  
  
    for (limite=n-1; limite>=1; limite--) {  
        max = limite ;  
        for (i=0; i<limite; i++) {  
            if (T[i] > T[max])  
                max = i ;  
        }  
        if (max != i) {  
            aux = T[i] ;  
            T[i] = T[max] ;  
            T[max] = aux ;  
            system ("CLS") ; afficheTab (T, n) ;  
            printf ("\n\n") ; system ("PAUSE") ;  
        }  
    }  
}
```

Des tris plus efficaces

- ◆ Le tri à bulle et le tri par sélection sont très souvent utilisés
 - Très facile à implémenter
 - MAIS très mauvais en terme de complexité
- ◆ D'autres tris plus efficaces
 - Tri par fusion
 - Quick sort

Supprimer une valeur

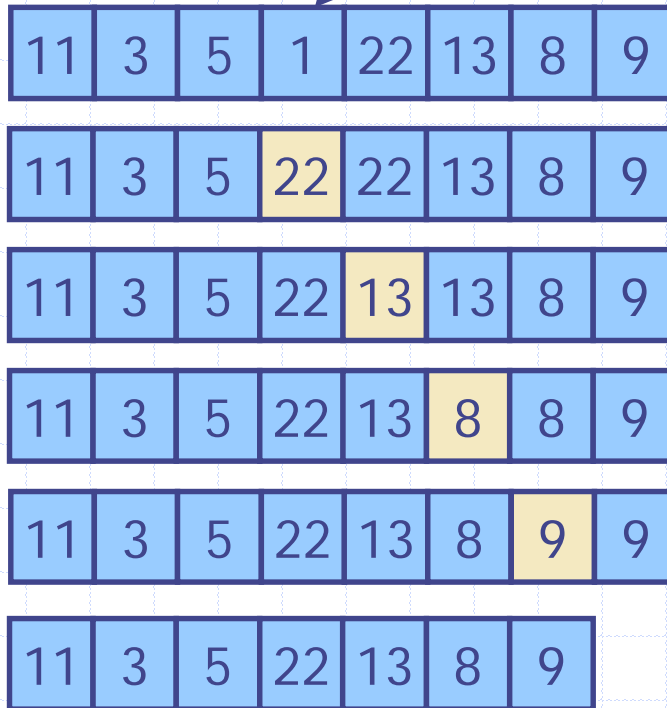
◆ 2 possibilités

- Suppression d'une valeur à un indice donné
 - ◆ Cas le plus facile
 - On effectue des décalages vers la gauche à partir de l'indice donné
 - On diminue la taille du tableau
- Suppression d'une valeur donnée si elle est présente dans le tableau
 - ◆ Plus complexe
 - On recherche la valeur à supprimer
 - À partir de l'emplacement de cette valeur, on effectue des décalages vers la gauche jusqu'à la fin du tableau
 - On diminue la taille du tableau

Supprimer une valeur

◆ Suppression d'une valeur donnée

Suppression de la case 3



Insertion d'une valeur

- ◆ Première question à se poser: reste-t-il de la place dans le tableau?
- ◆ Erreur souvent commise : partir du début du tableau
- ◆ 2 possibilités
 - Ajout d'une valeur à un indice donné
 - ◆ À partir de la fin du tableau et jusqu'à l'indice donné on fait des décalages vers la droite
 - ◆ On augmente la taille du tableau
 - Ajout d'une valeur donnée
 - ◆ On considère dans ce cas que le tableau est trié
 - ◆ À partir de la fin du tableau et jusqu'à la valeur donnée on fait des décalages vers la droite
 - ◆ On augmente la taille du tableau

Insertion d'une valeur

