

LOGIQUE / PROGRAMMATION C

TRAVAUX DIRIGES

Les techniques de recherche dans les ensembles non triés



Bloc 1 – Informatique de gestion

Techniques de recherche

1. Les vecteurs : rappels

Jusqu'à présent, on a travaillé avec des variables de type « simple », mais on peut définir des variables + « complexes » :

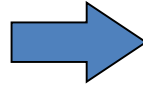


Tableau =

ensemble de variables de MEME TYPE, stockées en mémoire à des adresses contigües et qui sont regroupées sous un même identificateur

Techniques de recherche

1. Les vecteurs : rappels

- **Déclaration :**

La taille et le type d'un tableau doivent être déclarés

| | |
|---------------------------------|---------------------------------|
| <code>float vec[3] ;</code> | Tableau à 1 dimension = vecteur |
| <code>char tab [10][4] ;</code> | Tableau à 2 dimensions |

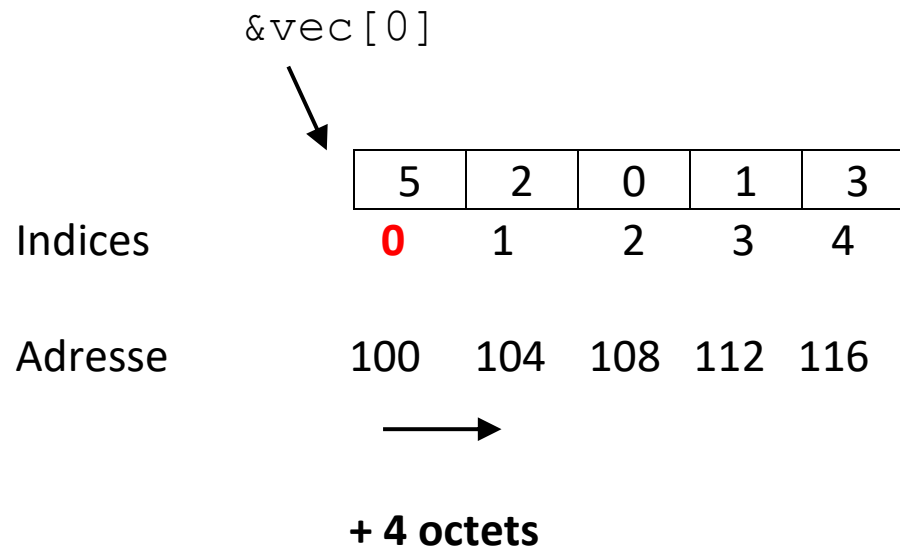
Cette déclaration alloue donc en mémoire pour l'objet *tab* un espace de 10×4 octets consécutifs.

Techniques de recherche

1. Les vecteurs : rappels

- **Implantation en mémoire :**

```
int vec[5] ;
```



Techniques de recherche

1. Les vecteurs : rappels

- **Initialisation :**

On peut initialiser un tableau de 2 manières :

- lors de sa déclaration par une liste de constantes :

```
type nom-du-tableau[N] = {constante-1,constante-2,...,constante-N};
```

Par exemple, on peut écrire

```
#define N 4
.....
int tab[N] = {1, 2, 3, 4};
```

- par des affectations aux éléments :

```
int vec [10] ;
vec[0]=12 ; vec[1]=8 ; vec[2]=23 ;... vec[9]=4 ;
```

Techniques de recherche

1. Les vecteurs : rappels

- **Comment accéder aux éléments ?**

On accède à un élément du tableau en lui appliquant l'opérateur [], par l'intermédiaire de son indice. Les éléments d'un tableau sont toujours numérotés de 0 à nombre_élément -1.

Ex : `vec[0] = 5, vec[1] = 2, ...`

Techniques de recherche

1. Les vecteurs : rappels

Remarques :

L'indice :

- *un nombre, une variable ou une variable, ou une expression calculée*
- *≥ 0 , integer et inférieur au nombre d'éléments du tableau.*

```
float vec [10];  
int i;
```

```
/....
```

```
    vec[i]=5;  
    vec[2*i]=-2;
```

```
    vec[10] = 3;
```

```
.../
```

Techniques de recherche

1. Les vecteurs : rappels

- **Comment remplir un vecteur ?**

```
#define N 10

int main(void)
{
    int tab[N];
    int i;
    ...
    for (i = 0; i < N; i++)
        scanf("%d", &tab[i]);
}
```


Techniques de recherche

2. *Les algorithmes*

- **Introduction :**

Un problème courant en algorithmique :

-> détecter la présence ou non d'un élément dans un ensemble de données (vecteur, un tableau, un fichier,...)

2 cas de figure peuvent se présenter :

- **L'ensemble des données est trié**
- **L'ensemble des données est non trié**

Techniques de recherche

2. Les algorithmes

- **Introduction :**

Si les données ne sont pas triées, les différentes méthodes envisageables consistent toutes à ***parcourir le tableau de données du 1^{er} élément et à s'arrêter dès qu'on a trouvé l'élément cherché*** (on ne cherche pas toutes les occurrences de l'élément).

Cible = 7 ?

{ 1 , 3 , -3 , 4 , 7 , 9 , 0 , - 2 , 7 , 4 }

= méthodes de recherche séquentielle (ou linéaire).

Techniques de recherche

2. Les algorithmes

Question : quid des performances ??

Pas vraiment optimal ☹ :

Pour un tableau de N éléments, soit on tombe sur le bon élément après 1 opération...mais au pire des cas on devra effectuer N opérations ...la probabilité moyenne de trouver l'élément est donc de $p=N/2$.

Techniques de recherche

2. *Les algorithmes*

4 techniques :

- a) Technique classique
- b) Technique du booléen
- c) Technique du forçage de boucle
- d) Technique de la sentinelle

Techniques de recherche

2. *Les algorithmes*

a) Technique classique :

```
int vec[50], cible;
```

```
printf(" recherche de quel nombre ? ");  
scanf("%d",&cible);
```

| | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|
| 3 | -1 | 6 | 9 | 3 | 8 | 7 | 0 | 7 | 4 |
|---|----|---|---|---|---|---|---|---|---|

Parcourir du 1^{er} élément au dernier :

- Si la cible est trouvée, on arrête le parcours , on affiche « cible trouvée » **et sa position dans le tableau**
- Sinon, « cible non trouvée »

Techniques de recherche

2. Les algorithmes

a) Technique classique :

Si cible = 7 :

Quitter la recherche -> élément trouvé



| | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|
| 3 | -1 | 6 | 9 | 3 | 8 | 7 | 0 | 7 | 4 |
|---|----|---|---|---|---|---|---|---|---|



Parcours du 1^{er} élément au 1^{er} élément = 7

Techniques de recherche

2. *Les algorithmes*

b) Technique du booléen :


Dans cette technique, on utilise une variable booléenne **trouve** qui passera de 0 à 1 si la cible est présente et a été trouvée dans le tableau.

Si cible = 7 :

| | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|
| 3 | -1 | 6 | 9 | 3 | 8 | 7 | 0 | 7 | 4 |
|---|----|---|---|---|---|---|---|---|---|



Au départ, *trouve*=0



Si cible trouvée, *trouve*=1 : on sort de la boucle

Techniques de recherche

2. Les algorithmes

c) Technique du forçage de boucle :

Dans cette technique, on suppose que la recherche va se faire dans tout le vecteur ($i < n$), la boucle s'arrêtera lorsqu'on aura trouvé la cible en donnant à i une valeur plus grande que n .

Si cible = 7 :

| | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|
| 3 | -1 | 6 | 9 | 3 | 8 | 7 | 0 | 7 | 4 |
|---|----|---|---|---|---|---|---|---|---|

Parcours de tout le vecteur de $i=0$ jusque n


Si cible trouvée, $i=n$: on sort de la boucle

Techniques de recherche


2. Les algorithmes

c) Technique du forçage de boucle :

| | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|
| 3 | -1 | 6 | 9 | 3 | 8 | 7 | 0 | 7 | 4 |
|---|----|---|---|---|---|---|---|---|---|



Parcours de tout le vecteur de $i=0$ jusque n



Si cible trouvée, $i=n$: on sort de la boucle

Attention : à la sortie de la boucle, besoin d'une condition pour savoir si la cible a été trouvée ou non ??

Techniques de recherche

2. *Les algorithmes*

d) Technique de la sentinelle :

Cette technique consiste à placer la valeur de la cible après la dernière donnée de l'ensemble dans lequel s'effectue la recherche.

Dans un tableau contenant n informations, on place la valeur cible dans la case $n+1$. Il est évident que dans ce cas, **toute recherche va aboutir.**

On pourra conclure sur la présence de la cible dans le vecteur si on a trouvé la cible avant la case $n+1$.

Techniques de recherche

2. Les algorithmes

d) Technique de la sentinelle :

Si cible = 2 :

Vecteur de départ :

| | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|
| 3 | -1 | 6 | 9 | 3 | 8 | 7 | 0 | 7 | 4 |
|---|----|---|---|---|---|---|---|---|---|

Au début de la recherche:

| | | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|---|
| 3 | -1 | 6 | 9 | 3 | 8 | 7 | 0 | 7 | 4 | 2 |
|---|----|---|---|---|---|---|---|---|---|---|



Parcours de tout le
vecteur de $i=0$ jusqu'à $n+1$

Techniques de recherche

2. *Les algorithmes*

A vous de jouer....

- Programmer en C les 4 méthodes
- Sur papier
- Faire photo + envoyer 1 fichier d'une feuille reprenant les 4 algorithmes...
- Renommer fichier (.jpg) avec votre nom+ prénom et envoyer via messagerie EV

