



KỸ THUẬT LẬP TRÌNH



BÀI 4

KỸ THUẬT LẬP TRÌNH VỚI MẢNG



BÀI 4. KỸ THUẬT LẬP TRÌNH VỚI MẢNG

- 1. Khái niệm mảng**
- 2. Các thao tác cơ bản trên mảng**
- 3. Các bài toán cơ bản**
 - Bài toán tìm kiếm
 - Bài toán sắp xếp
 - Các bài toán khác
- 4. Mảng hai chiều**



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

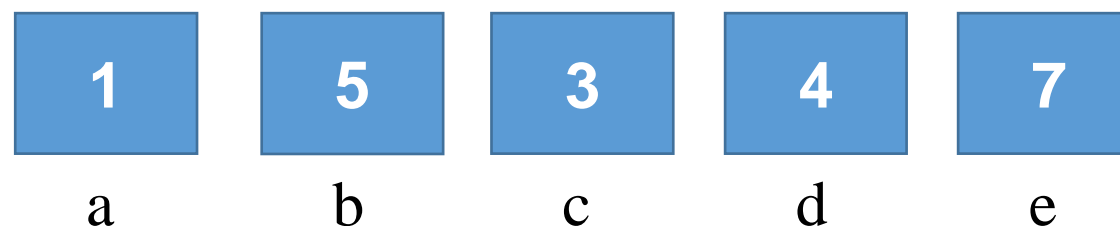
Bài toán khác

Mảng hai chiều

- **Khái niệm mảng**

- **Bài toán:** Hãy tổ chức lưu trữ $n = 5$ giá trị nguyên sau:
1, 5, 3, 4, 7

Cách 1: Sử dụng 5 biến nguyên a, b, c, d, e



👉 **Hãy chỉ ra nhược điểm của cách này !**



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

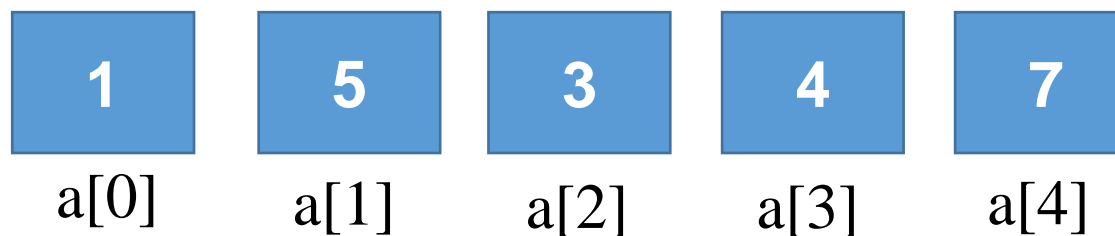
Bài toán khác

Mảng hai chiều

- **Khái niệm mảng**

- **Bài toán:** Hãy tổ chức lưu trữ 5 giá trị nguyên sau:
1, 5, 3, 4, 7

Cách 2: Sử dụng một cấu trúc dữ liệu gồm 5 ô nhớ cùng kiểu nguyên, cùng tên nhưng khác nhau về chỉ số:



☞ Cấu trúc dữ liệu trên được gọi là mảng a (array a)



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Khái niệm mảng**

- **Mảng (mảng một chiều)** là một cấu trúc dữ liệu gồm các ô nhớ liên tiếp cùng tên, cùng kiểu nhưng khác nhau về chỉ số.
- **Chỉ số của mảng (index):** là các số nguyên bắt đầu từ 0.

phần tử của mảng



a[0] a[1] a[2] a[3] a[4]

1 5 3 4 7

0 1 2 3 4

chỉ số của mảng





Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Khái niệm mảng**

- ✍ **Mảng:** Dùng để lưu trữ một dãy các giá trị cùng kiểu.

- ✍ **Trong thực tế, cấu trúc dữ liệu mảng rất phổ biến.**

- Có trong hầu hết các ngôn ngữ lập trình.
 - Lưu trữ và cho phép thao tác dễ dàng trên một tập giá trị cùng kiểu như: Họ tên, Lương, Tuổi, Chiều cao, Cân nặng, ...



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Các thao tác cơ bản trên mảng**

- **Khai báo mảng**
- **Nhập mảng**
- **Xuất mảng**
- **Duyệt mảng**



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Các thao tác cơ bản trên mảng**

- **Khai báo mảng**

⟨Kiểu⟩ ⟨Tên⟩ [⟨p⟩];

⟨Kiểu⟩: Là kiểu dữ liệu của tất cả các phần tử trong mảng

⟨Tên⟩: Tên mảng tùy ý đặt, tuân theo quy tắc đặt tên.

⟨p⟩: Là một số nguyên, chỉ ra kích thước tối đa của mảng.

Ví dụ: **int a[5];**

→ Ta có một mảng a có tối đa 5 phần tử kiểu nguyên



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Các thao tác cơ bản trên mảng**

- **Nhập dữ liệu cho mảng**

- Dữ liệu của mảng có thể được khởi tạo:

Ví dụ: float a[5] = {1.1, 2.2, 3.0, 1.8, 6.4};

- Dữ liệu của mảng có thể được nhập từ bàn phím:

Ví dụ: đoạn chương trình sau cho phép nhập **n** (là kích thước thực tế của mảng) và sử dụng vòng lặp **for** để nhập vào từ bàn phím một mảng **a** gồm **n** phần tử kiểu số.

```
cout<<"n="; cin>>n;
for(int i=0; i<n; i++)
{
    cout<<"a["<<i<<"]=";
    cin>>a[i];
}
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Các thao tác cơ bản trên mảng**

- **Xuất dữ liệu của mảng lên màn hình**

- Ta sử dụng vòng lặp **for** để xuất lần lượt từng phần tử của mảng lên màn hình bằng lệnh **cout**
 - Đoạn chương trình sau xuất dữ liệu của một mảng **a** gồm **n** phần tử lên màn hình:

```
for(int i=0; i<n; i++)  
    cout<<a[i];
```

- Ta có thể trang trí một chút:

```
for(int i=0; i<n; i++)  
    cout<<setw(5)<<a[i];
```

```
for(int i=0; i<n; i++)  
    cout<<a[i]<<" ";
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Các thao tác cơ bản trên mảng**

- **Duyệt mảng**

- Duyệt mảng là thao tác ta “thăm” lần lượt các phần tử của mảng từ trái qua phải hoặc ngược lại.
 - Sử dụng vòng lặp **for** cho duyệt mảng:

Duyệt xuôi

```
for(int i=0; i<n; i++)  
    //Thăm a[i];
```

Duyệt ngược

```
for(int i=n-1; i>=0; i--)  
    //Thăm a[i];
```

✍ Duyệt mảng là thao tác phổ biến nhất khi xử lý mảng !



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

BÀI TẬP 4.1

- Nhập vào từ bàn phím một mảng a gồm n phần tử số thực. Hãy:
 - Tìm và in ra giá trị lớn nhất trong mảng.
 - Cho biết mảng có bao nhiêu giá trị lớn nhất



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán tìm kiếm**

- Cho một mảng a gồm n phần tử và một phần tử C cùng kiểu. Hỏi C có xuất hiện trong a hay không.

- **Phương pháp tìm kiếm tuần tự:**

Duyệt mảng và đếm số lần xuất hiện của C trong mảng

```
int d = 0;
for(int i=0; i<n; i++)
    if(a[i] == C)
        d++;
```

Biến d chính là số lần xuất hiện của C trong mảng a .



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán tìm kiếm**

- **Phương pháp tìm kiếm tuần tự:**

- Căn cứ vào giá trị của d sau khi DUYỆT XONG MẢNG, ta có thể kết luận YES/ NO:

```
int d = 0;
for(int i=0; i<n; i++)
    if(a[i] == C)
        d++;
if(d == 0)
    cout<<"NO";
else
    cout<<"YES";
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán tìm kiếm**

- **Phương pháp tìm kiếm tuần tự:**

- **Dừng lại sớm:** Nếu trong quá trình duyệt ta gặp C, khi đó ta có thể dừng tìm kiếm và trả lời Yes.
 - **Hàm tìm kiếm tuần tự:** Trả về **true** nếu tìm thấy C trong một mảng a gồm n số thực. Ngược lại hàm trả về **false**:

```
bool Find(float a[], int n, float C)
{
    for(int i=0; i<n; i++)
        if(a[i] == C)
            return true;
    return false;
}
```




Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều



BÀI TẬP 4.2

- Nhập vào từ bàn phím một mảng a gồm n phần tử nguyên. Tính tổng của các phần tử vừa chẵn, vừa lớn hơn 10 trong mảng vừa nhập.



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán tìm kiếm**

- **Tìm kiếm nhị phân trên mảng được sắp**

- Nếu mảng a đã được sắp tăng hoặc giảm, ta có thể có một giải thuật tìm kiếm nhanh hơn.
- **Ví dụ:** Mảng a sau gồm n=9 phần tử và C = 10.

1	3	4	5	7	8	9	10	12
---	---	---	---	---	---	---	----	----

C=10

- Nếu tìm kiếm tuần tự, trường hợp tồi nhất ta cần duyệt hết mảng. Độ phức tạp tính toán $O(n)$
- Có hay không một thuật toán tìm kiếm chỉ thăm 3 phần tử của mảng có thể trả lời YES/NO ?



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

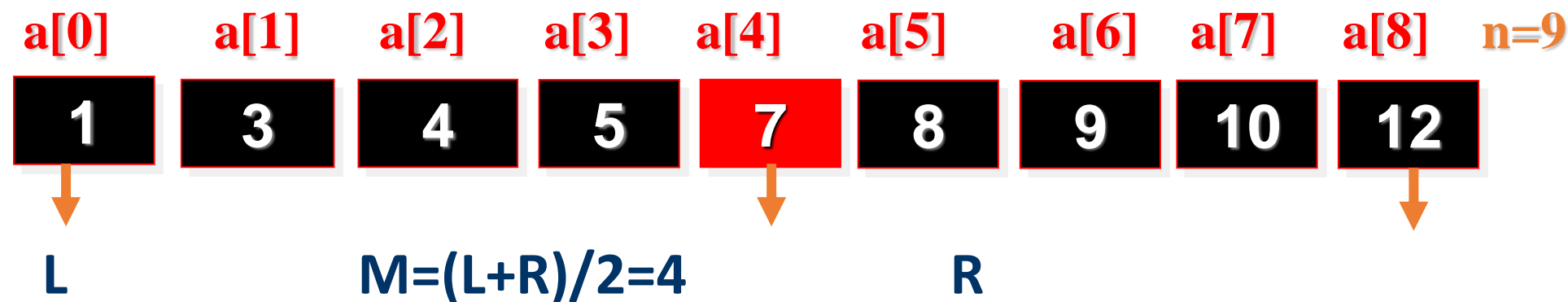
Bài toán khác

Mảng hai chiều

- Bài toán tìm kiếm

- Tìm kiếm nhị phân trên mảng được sắp

C=10



$a[M] = c$: Yes

$L > R$: No

$a[M] < c$: $L=M+1$;

$a[M] > c$: $R=M-1$;



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán tìm kiếm**

- **Tìm kiếm nhị phân trên mảng được sắp**

Hàm sau trả về -1 nếu không tìm thấy C, ngược lại, hàm trả về 1 vị trí của C trong mảng a.

```
int TKNP_DQ(int a[], int c, int L, int R)
{
    int M=(L+R)/2;
    if(a[M]==c)           return M;
    else if(L>R)           return -1;
    else
        if(a[M]>c)         return TKNP_DQ(a,c,L,M-1);
        else               return TKNP_DQ(a,c,M+1,R);
}
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- Bài toán tìm kiếm

- Tìm kiếm nhị phân trên mảng được sắp

```
void TKNP_Lap(int a[], int n, int c)
{
    int L=0, R=n-1, M;
    do
    { M = (L+R) / 2;
      if (a[M]>c) R = M-1;
      if (a[M]<c) L = M+1;
    }
    while (a[M] != c && L<R);
    if (a[M]==c) cout<<c<<" xuất hiện tại "<<M;
    else        cout<<c<<" không xuất hiện";
}
```

Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

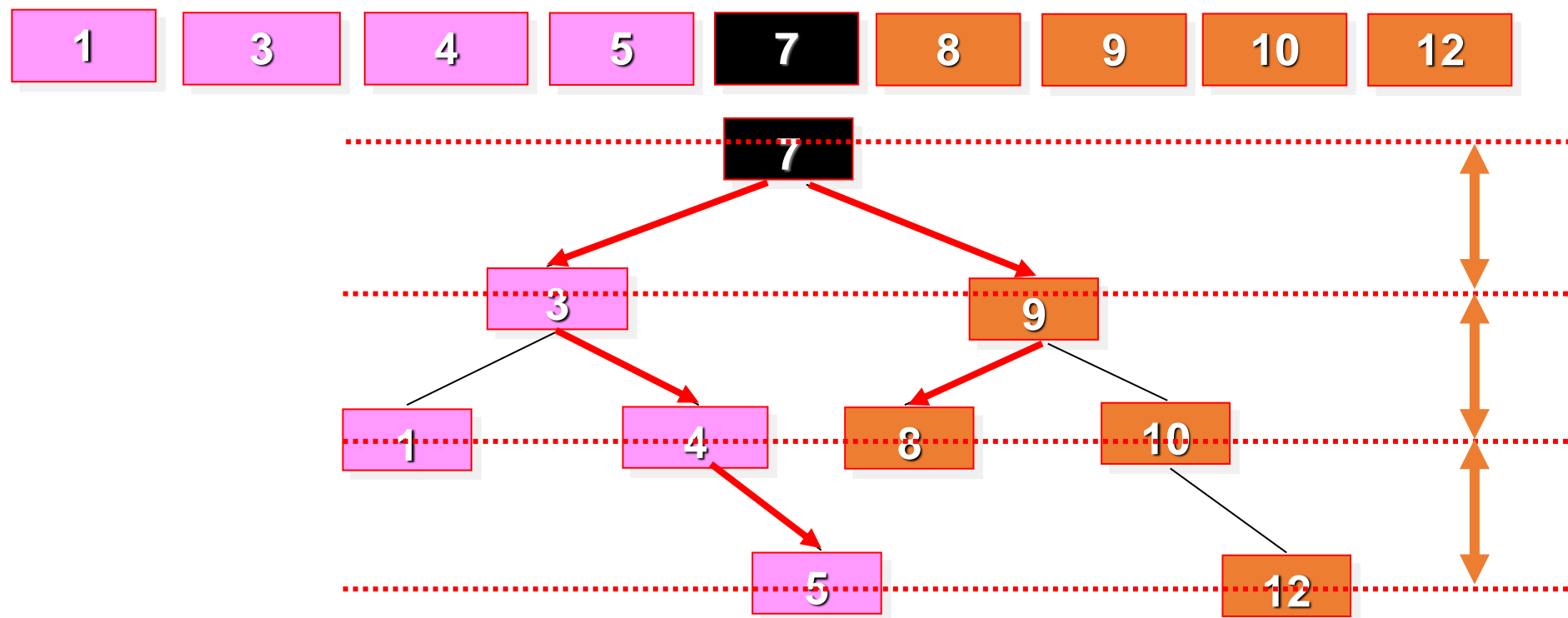
Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- Bài toán tìm kiếm

- Tìm kiếm nhị phân trên mảng được sắp



☞ Chiều cao của cây: $\log_2(n)$;

☞ Trường hợp tồi nhất: Đi hết chiều cao của cây, nên độ phức tạp tính toán là $O(\log_2(n))$



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

BÀI TẬP 4.3

- Nhập vào từ bàn phím một mảng a gồm n phần tử nguyên sao cho các phần tử của mảng được sắp tăng và một phần tử nguyên C .
- Hãy cho biết C xuất hiện trong mảng hay không (mà không duyệt mảng).



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán sắp xếp**

- Cho một mảng a gồm n phần tử. Hãy sắp a tăng dần

- **Phương pháp sắp nổi bọt**

- Sắp lần lượt từng phần tử, từ trái qua
- Để sắp phần tử thứ i ta:
 - ✓ Duyệt qua các phần tử đứng sau nó
 - ✓ Nếu gặp phần tử nào nhỏ hơn nó thì đổi chỗ
- Thao tác đổi chỗ $a[i]$ và $a[j]$:

```
Tg = a[i];  
a[i] = a[j];  
a[j] = Tg;
```




Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán sắp xếp**

- Cho một mảng a gồm n phần tử. Hãy sắp a tăng dần
- **Phương pháp sắp nổi bọt (Bubble Sort)**

```
void BubbleSort(float a[], int n)
{
    for(int i = 0; i < n; i++)
        for(int j = i+1; j < n; j++)
            if(a[j] < a[i])
            {
                float Tg = a[i];
                a[i] = a[j];
                a[j] = Tg;
            }
}
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán sắp xếp**

- **Phương pháp sắp chọn (Selection Sort)**

- Sắp lần lượt từng phần tử, từ trái qua
 - Để sắp phần tử thứ i ta:
 - ✓ Tìm phần tử nhỏ nhất trong số các phần tử đứng sau nó, kể cả nó
 - ✓ Đổi chỗ phần tử đang sắp và phần tử nhỏ nhất tìm được

- Thao tác tìm min: tìm vị trí min thay vì tìm giá trị min.



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán sắp xếp**

- **Phương pháp sắp chọn (Selection Sort)**

```
void SelectionSort(float a[], int n)
{
    for(int i = 0; i < n; i++)
    {
        int Min = i;
        for(int j = i+1; j<n; j++)
            if(a[j] < a[Min]) Min = j;
        tg = a[i];
        a[i] = a[Min];
        a[Min] = tg;
    }
}
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán sắp xếp**

- **Phương pháp sắp chèn (Insertion Sort)**

- Nếu mảng a chỉ có 1 phần tử, hiển nhiên nó đã được sắp
 - $a[0]$ luôn được coi là đã được sắp.
 - Duyệt qua các phần tử từ phần tử $a[1]$ về sau
 - Với mỗi phần tử, tìm vị trí và chèn vào đoạn mảng đã được sắp sao cho không phá vỡ tính được sắp của mảng.



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán sắp xếp**

- **Phương pháp sắp chèn (Insertion Sort)**

- Duyệt mảng từ $a[1]$ tới hết
 - Với mỗi phần tử được duyệt $a[i]$, ta:
 - Đặt $a[i]$ ra biến Tg
 - Duyệt qua các phần tử đứng trước $a[i]$
 - Chừng nào chưa duyệt hết mảng và gặp một phần tử lớn hơn Tg thì đẩy phần tử đó sang phải.
 - Chèn Tg vào ngay sau vị trí duyệt



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán sắp xếp**

- **Phương pháp sắp chèn (Insertion Sort)**

```
1 void SapChen(int a[], int n)
2 {
3     for (int i=1; i< n; i++)
4     {
5         int Tg = a[i];    int j = i-1;
6         while (j >= 0 && a[j] > Tg)
7         {
8             a[j+1] = a[j];    j--;
9         }
10        a[j+1] = Tg;
11    }
12 }
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều



BÀI TẬP 4.4

- Nhập vào một mảng a gồm n phần tử nguyên. Sắp a tăng dần bằng một trong ba phương pháp đã học. In kết quả ra màn hình.



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Bài toán thống kê**

- Một dãy số a gọi là
 - Được sắp “tăng” nếu $a[i] \leq a[i+1]$;
 - Được sắp “giảm” nếu $a[i] \geq a[i+1]$;
 - Được sắp “tăng ngặt” nếu $a[i] < a[i+1]$ và
 - Được sắp “giảm ngặt” nếu $a[i] > a[i+1]$, $\forall i \in [0, n-1]$.

Viết chương trình nhập một dãy n số thực, kiểm tra xem dãy đã được sắp hay chưa. Nếu đã được sắp thì sắp theo trật tự nào (tăng, tăng ngặt, giảm, giảm ngặt).



Khái niệm

Thao tác cơ bản

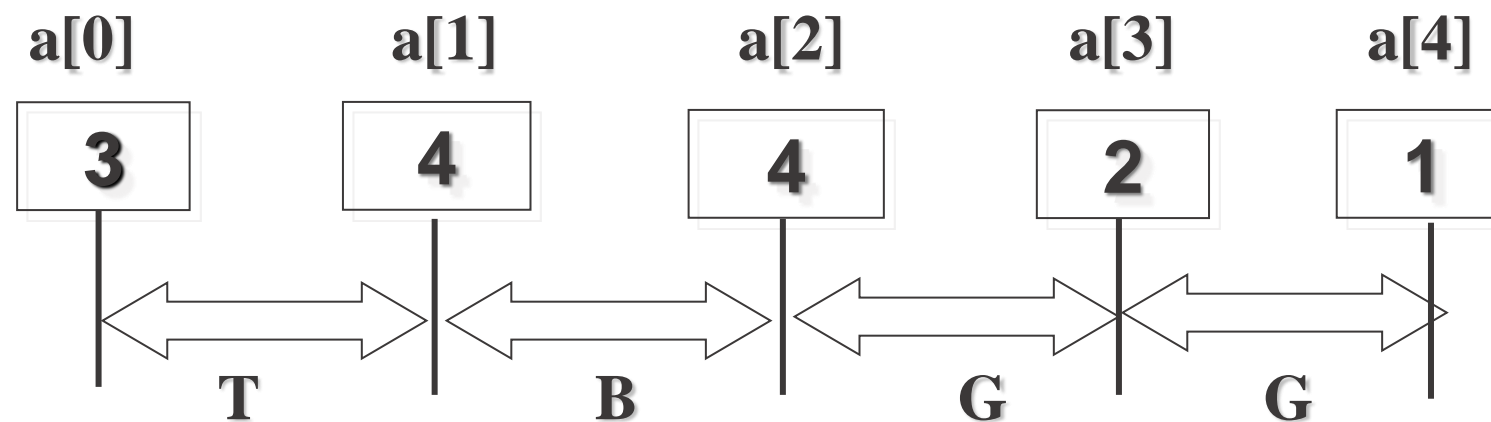
Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- Bài toán thống kê



Bước 1: Duyệt và đếm T, G, B

Bước 2: Căn cứ vào T, G, B để kết luận



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- Bài toán thống kê

```
void CheckSort(float a[], int n)
{
    int T, G, B ;   T = G = B = 0;
    for(int i=0 ; i < n-1 ; i++)
    {
        if (a[i] < a[i+1])           T++;
        if (a[i] > a[i+1])           G++;
        if (a[i] == a[i+1])          B++;
    }
    if (B==n-1)           cout<< "Mang gom cac phan tu bang nhau";
    else if (T == n-1)    cout<< "Mang duong sap tang ngat";
    else if (G == n-1)    cout<< "Mang duoc sap giam ngat";
    else if (T == 0 && B != 0)    cout<< "Mang duoc sap giam";
    else if (G == 0 && B != 0)    cout<< "Mang duoc sap tang";
    else                   cout<< "Mang chua duoc sap";
}
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều



BÀI TẬP 4.5

- Trong một hàng cây, người ta đo chiều cao của từng cây và lưu trữ trong một mảng a . Một hàng cây được gọi là “khá đồng đều” nếu có ít nhất 3 cây liên tiếp có độ cao bằng nhau. Hãy xác định xem một hàng cây có thể gọi là khá đồng đều hay không.



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Khái niệm**

- **Bài toán:** hãy lưu trữ bảng số sau

1	2	3	4
3	2	4	3
1	4	2	5

- **Cách 1:** dùng 3 mảng 1 chiều:

a	→	1	2	3	4
b	→	3	2	4	3
c	→	1	4	2	5



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Khái niệm**

- **Cách 2:** dùng cấu trúc dữ liệu mảng hai chiều:

a	0	1	2	3	j	chỉ số cột
0	1	2	3	4		
1	3	2	4	3		
2	1	4	2	5		

chỉ số dòng **i**

- Mảng hai chiều là một cấu trúc dữ liệu dùng để lưu trữ một bảng các giá trị cùng kiểu
- Chỉ số của các phần tử mảng gồm: Chỉ số dòng và chỉ số cột



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

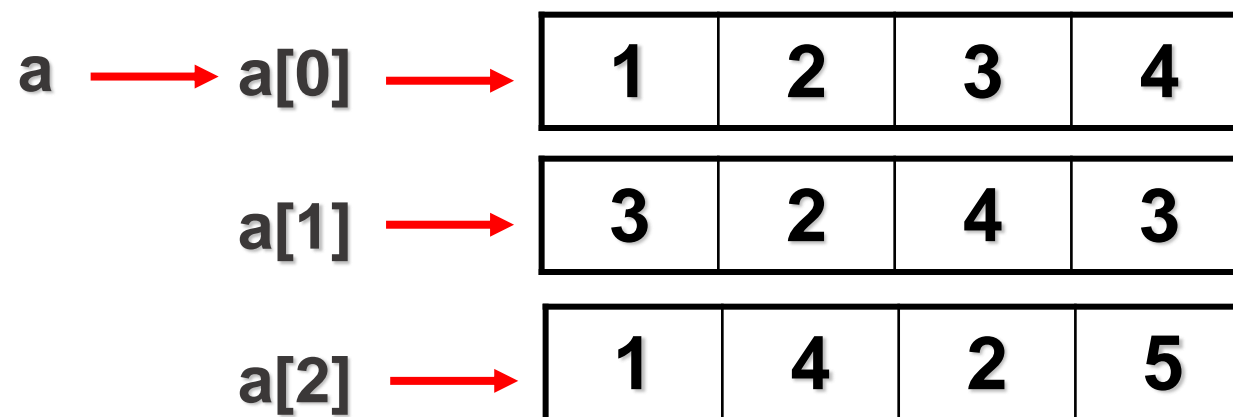
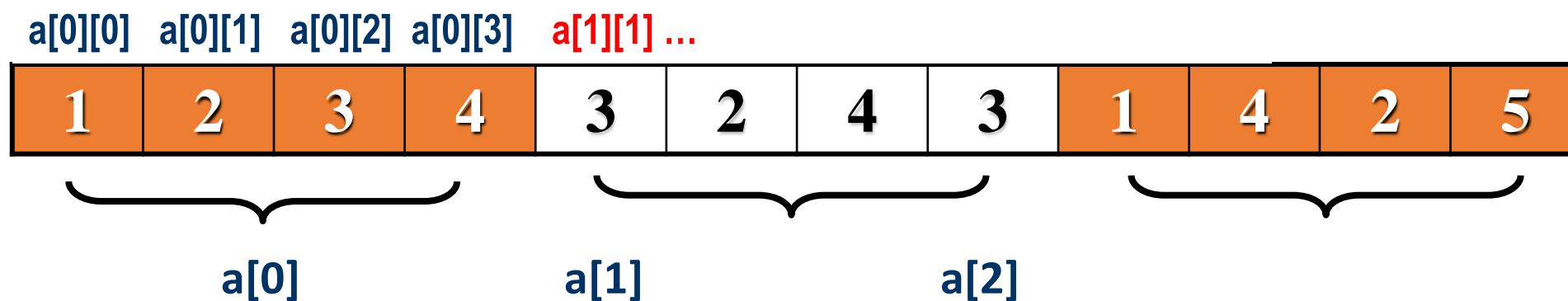
Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Khái niệm**

- Mảng hai chiều thực chất là mảng của các mảng một chiều:





Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mạng hai chiều

- **Khái niệm**

- Cấu trúc mạng hai chiều cũng được sử dụng khá phổ biến. Đặc biệt rất phù hợp cho một số loại dữ liệu:
 - Dữ liệu là các ảnh số
 - Dữ liệu đồ thị
 - Các hệ thống trí tuệ nhân tạo sử dụng tính toán tensor như: Mạng CNN
 - Biểu diễn các bài toán tối ưu trong máy tính
 - ...



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Các thao tác cơ bản**

- [1]. Khai báo mảng**

⟨Kiểu⟩ ⟨Tên⟩ [⟨p⟩][⟨q⟩];

Ví dụ: int a[100][100];

- Khai báo một mảng a gồm 100x100 phần tử kiểu nguyên.
- Mảng a có thể chứa một bảng số nguyên có tối đa 100 dòng và 100 cột.



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Các thao tác cơ bản**

- **[2]. Nhập mảng**

- Sử dụng hai vòng lặp for lồng nhau để nhập mảng

```
void nhapmang(float a[100][100], int &n, int &m)
{
    cout<<"n="; cin>>n;
    cout<<"m="; cin>>m;
    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++)
        {
            cout<<"a["<<i<<"]["<<j<<"]="";
            cin>>a[i][j];
        }
}
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Các thao tác cơ bản**

- **[3]. Xuất mảng ra màn hình**

- Sử dụng hai vòng lặp for lồng nhau để xuất mảng

```
void xuatmang(float a[100][100], int n, int m)
{
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<m; j++)
            cout<<setw(5)<<a[i][j];
        cout<<endl;
    }
}
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Các thao tác cơ bản**

- **[4]. Duyệt mảng**

- Sử dụng hai vòng lặp for lồng nhau để duyệt mảng

```
for(int i=0; i<n; i++)  
for(int j=0; j<m; j++)  
    //Thăm a[i][j]
```



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

BÀI TẬP 4.6

- Nhập vào một ma trận gồm n dòng, m cột các số thực. In ma trận vừa nhập ra màn hình.
- Cho biết giá trị lớn nhất trong ma trận



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

- **Một số dạng bài toán trên mảng hai chiều**

- Tìm kiếm trên mảng
- Thống kê trên mảng, kiểm tra tính chất của mảng
- Các tính toán ma trận: Cộng, trừ, nhân hai ma trận, chuyển vị, đổi dấu ma trận,...

☞ **Phương pháp chung: Duyệt mảng**



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

BÀI TẬP 4.7

- Có n cửa hàng kinh doanh trong m tháng. Doanh thu của mỗi cửa hàng trong mỗi tháng đều được lưu trữ trong một ma trận có n dòng, m cột.
- Một cửa hàng sẽ bị đóng cửa nếu doanh thu của nó giảm liên tiếp trong $m-1$ tháng (trừ tháng đầu tiên).
- Hãy cho biết cửa hàng nào trong số n cửa hàng trên sẽ bị đóng cửa.



Khái niệm

Thao tác cơ bản

Bài toán tìm kiếm

Bài toán sắp xếp

Bài toán khác

Mảng hai chiều

BÀI TẬP 4.7

```
void CheckClosed(float a[100][100], int n, int m)
{
    for(int i=0; i<n; i++)
    {
        int d=0;
        for(int j=0; j<m-1; j++)
            if(a[i][j]>a[i][j+1]) d++;
        if (d==m-1)
            cout<<"Close the store: "<<i+1<<endl;
    }
}
```



KẾT THÚC