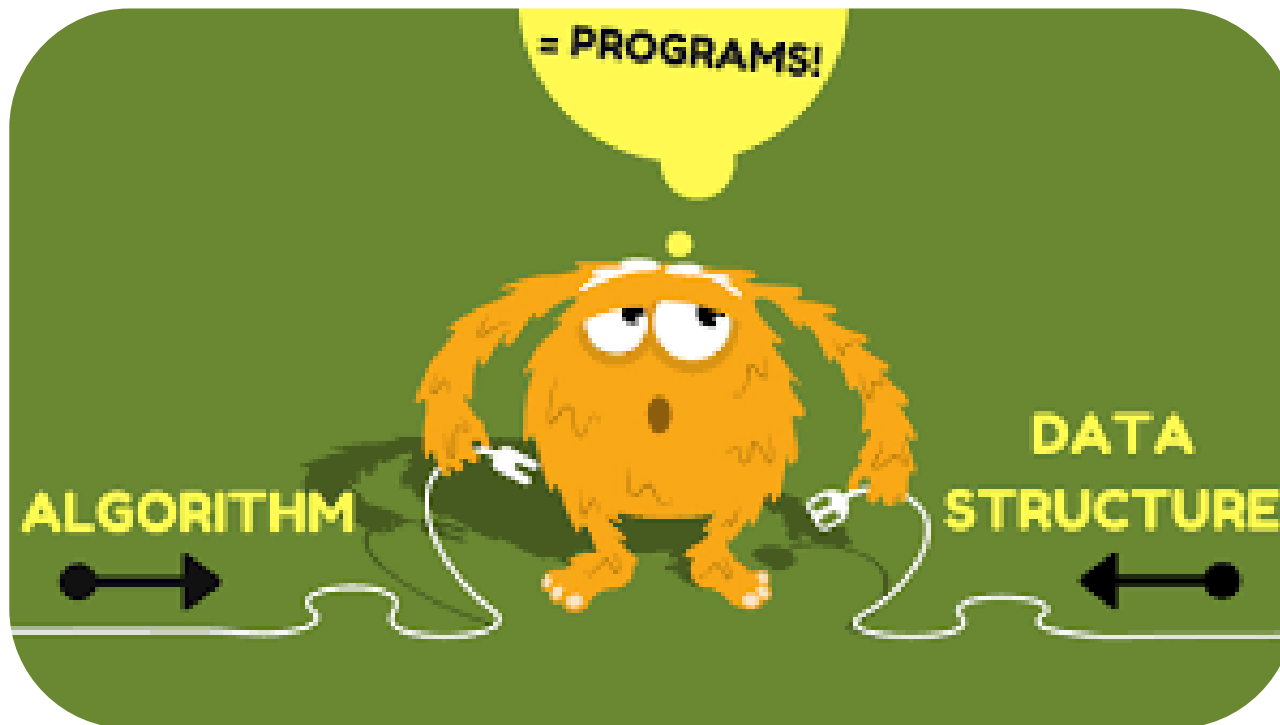


CÁC GIẢI THUẬT TÌM KIẾM



MỤC TIÊU BÀI HỌC

- Tìm hiểu các giải thuật tìm kiếm cơ bản trên cấu trúc dữ liệu mảng.
- Đánh giá và so sánh hiệu quả các giải thuật.



1

Giới thiệu

2

Tìm kiếm tuyến tính (tuần tự)

3

Tìm kiếm nhị phân

1

GIỚI THIỆU



- **Ý tưởng chung:** Từ khóa tìm kiếm **K**, ta chưa biết được vị trí của phần tử cần tìm, nên tiến hành so sánh K với lần lượt các phần tử trong dãy cần tìm cho đến khi ra kết quả (hoặc tìm thấy hoặc không tìm thấy).
- Có 2 loại giải thuật tìm kiếm theo cách này:
 - ✓ Tìm kiếm tuyến tính (tuần tự).
 - ✓ Tìm kiếm nhị phân.

2

TÌM KIẾM TUYẾN TÍNH



- Cho danh sách có n phần tử $a_0, a_1, a_2, \dots, a_{n-1}$.
- Để đơn giản trong việc trình bày giải thuật ta dùng mảng 1 chiều a để lưu danh sách các phần tử nói trên trong bộ nhớ chính.
- Tìm phần tử có khoá bằng X trong mảng
 - ✓ Giải thuật tìm kiếm tuyến tính (tìm tuần tự)
 - ✓ Giải thuật tìm kiếm nhị phân

➤ **Ý tưởng** : So sánh X lần lượt với phần tử thứ 1, thứ 2,...của mảng a cho đến khi gặp được khóa cần tìm, hoặc tìm hết mảng mà không thấy.

➤ **Các bước tiến hành**

Bước 1: Khởi gán $i=0$;

Bước 2: So sánh $a[i]$ với giá trị x cần tìm, có 2 khả năng

+ $a[i] == x$ tìm thấy x. Dừng;

+ $a[i] != x$ sang bước 3;

Bước 3: $i=i+1$ // Xét tiếp phần tử kế tiếp trong mảng Nếu $i==N$: dừng,

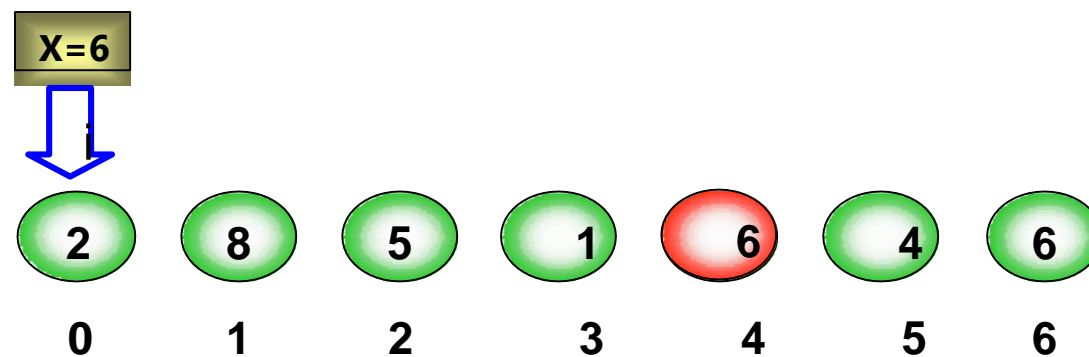
Ngược lại: Lặp lại bước 2;

- **Hàm trả về 1 nếu tìm thấy, ngược lại trả về 0:**

```
int LinearSearch(int a[],int n, int x)
{
    int i=0;
    while((i<n)&&(a[i]!=x))
        i++;
    if(i==n)
        return 0; //Tìm không thấy x
    else
        return 1; //Tìm thấy
}
```

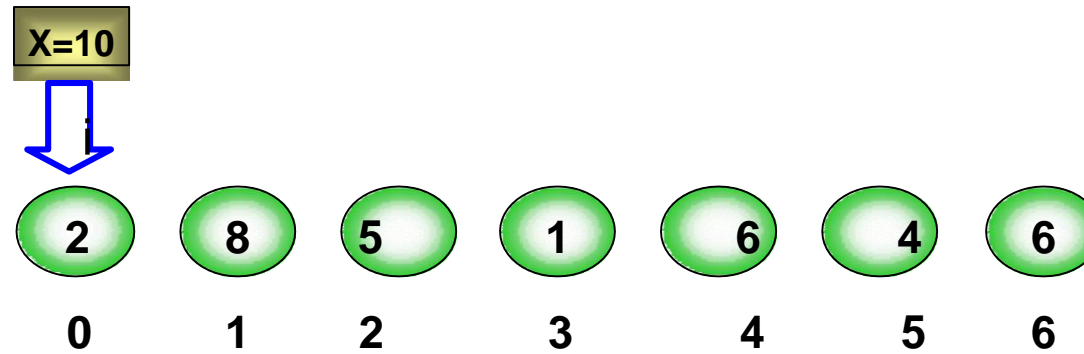
TÌM KIẾM TUYẾN TÍNH

➤ Mô tả thuật toán:



Tìm thấy 6 tại vị trí 4

➤ Mô tả thuật toán (tt):



$i=7$, không tìm thấy

➤ Độ phức tạp thuật toán:

Trường hợp	Css
Tốt nhất	1
Xấu nhất	$\cdot N$
Trung bình	$(N+1) / 2$

➤ Độ phức tạp $O(N)$

➤ Cải tiến thuật toán tìm tuyến tính:

- ✓ Nhận xét: Số phép so sánh của thuật toán trong trường hợp xấu nhất là **$2*n$** .
- ✓ Để giảm thiểu số phép so sánh trong vòng lặp cho thuật toán, thêm phần tử "lính canh" vào cuối dãy.

```
int LinearSearch(int a[], int n, int x)
{
    int i=0; a[n]=x; // a[n] là phần tử "lính canh"
    while(a[i]!=x)
        i++;
    if(i==n)
        return 0; // Tìm không thấy x
    else
        return 1; // Tìm thấy
}
```


3

TÌM KIẾM NHỊ PHÂN



THUẬT TOÁN TÌM KIẾM NHỊ PHÂN

➤ Được áp dụng trên mảng đã có thứ tự:

➤ Ý tưởng:

- ✓ Giả sử ta xét mảng có thứ tự tăng, khi ấy ta có $a_{i-1} < a_i < a_{i+1}$
- ✓ Nếu $X > a_i$ thì X chỉ có thể xuất hiện trong đoạn $[a_{i+1}, a_{n-1}]$
- ✓ Nếu $X < a_i$ thì X chỉ có thể xuất hiện trong đoạn $[a_0, a_{i-1}]$
- ✓ Ý tưởng mỗi bước so sánh X với phần tử đứng giữa trong dãy tìm kiếm hiện hành, dựa vào kết quả so sánh mà quyết định giới hạn dãy tìm kiếm ở nửa dưới hay nửa trên của dãy tìm kiếm hiện hành.

THUẬT TOÁN TÌM KIẾM NHỊ PHÂN

- Tìm kiếm hiện hành bao gồm các phần tử nằm trong a_{left} , a_{right}

Bước 1: $\text{left}=0$; $\text{right}=\text{N}-1$;

Bước 2: $\text{mid}=(\text{left}+\text{right})/2$; //chỉ số phần tử giữa dãy hiện hành

So sánh $a[\text{mid}]$ với x . Có 3 khả năng

$a[\text{mid}]=x$: tìm thấy. Dừng

$a[\text{mid}]>x$: $\text{Right}=\text{mid}-1$; $a[\text{mid}]<x$: $\text{Left}=\text{mid}+1$;

Bước 3: Nếu $\text{Left} \leq \text{Right}$; // còn phần tử trong dãy hiện hành

+ Lặp lại bước 2

Ngược lại : Dừng

CÀI ĐẶT THUẬT TOÁN TÌM KIẾM NHỊ PHÂN

- Hàm trả về 1 nếu tìm thấy, ngược lại hàm trả về giá trị 0

```
int BinarySearch(int a[],int n,int x)
{   int left, right, mid; left=0; right=n-1;
    do{
        mid=(left+right)/2;
        if(a[mid]==x) return mid;
        else if(a[mid]<x) left=mid+1;
        else right=mid-1;
    }while(left<=right);
    return 0;
}
```

ĐÁNH GIÁ THUẬT TOÁN KIỂM NHỊ PHÂN

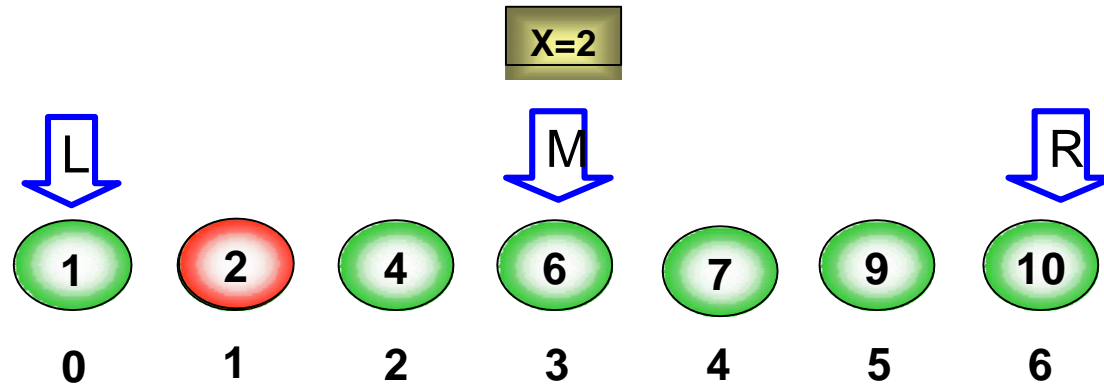
➤ Độ phức tạp của thuật toán

Trường hợp	Css
Tốt nhất	1
Xấu nhất	$\log_2 N$
Trung bình	$\log_2 N / 2$

➤ Độ phức tạp $O(\log_2 N)$

MINH HỌA THUẬT TOÁN TÌM KIẾM NHỊ PHÂN

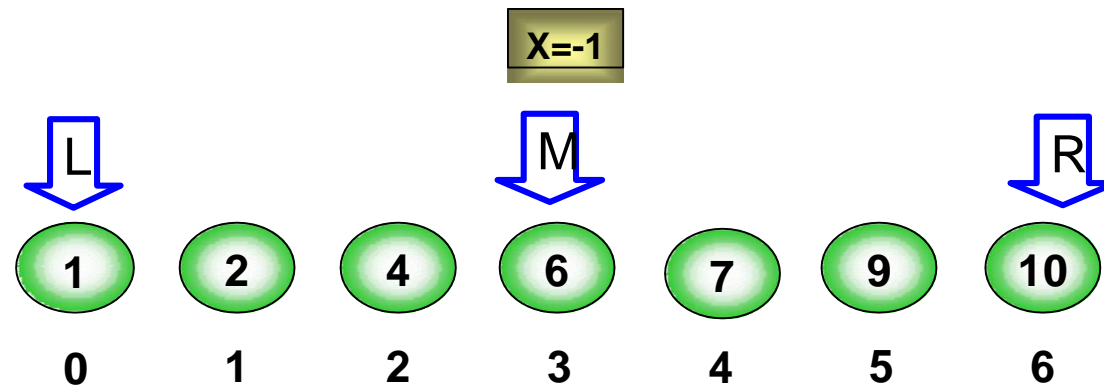
➤ Ví dụ 1:



Tìm thấy 2 tại vị trí 1

MINH HỌA THUẬT TOÁN TÌM KIẾM NHỊ PHÂN

➤ Ví dụ 2:



$L=0$

$R=-1 \Rightarrow$ không tìm thấy $X=-1$

- Sắp xếp và tìm kiếm là những bài toán quan trọng trong lập trình máy tính.
- Việc sắp xếp và tìm kiếm thường được tiến hành trên mảng.
- 2 giải thuật tìm kiếm với độ phức tạp $O(n)$:
 - ✓ Tìm kiếm tuyến tính: thực hiện trên mảng chưa được sắp thứ tự
 - ✓ Tìm kiếm nhị phân: thực hiện trên mảng đã được sắp thứ tự



THANK
YOU!

