

# Chương 01

## TÌM KIẾM TUYẾN TÍNH

### (LINEAR SEARCH)

TS. Nguyễn Tấn Trần Minh Khang

ThS. Cáp Phạm đình Thăng

Chương 01 - 1

# 1. THUẬT TOÁN

- Thuật toán **tìm tuyến tính** sẽ duyệt tất cả các đối tượng trên “**không gian tìm kiếm**” để tìm ra đối tượng thỏa mãn “**điều kiện tìm kiếm**”.
- Thông thường trong một bài toán kỹ thuật lập trình thì “**không gian tìm kiếm**” đơn giản nhất là: **mảng một chiều, ma trận, một đoạn giá trị nào đó, danh sách liên kết, cây,...**

TS. Nguyễn Tấn Trần Minh Khang

# 1. THUẬT TOÁN

- Mặt khác “**điều kiện tìm kiếm**” là tiêu chuẩn tìm kiếm được trình bày dưới dạng **một phát biểu không hình thức** và người lập trình phải **hình thức hóa nó bằng một “biểu thức logic”** trong chương trình.

TS. Nguyễn Tấn Trần Minh Khang

ThS. Cáp Phạm đình Thăng

Chương 01 - 3

## 2. ĐẶT ĐIỂM THUẬT TOÁN

- Thuật toán tìm kiếm tuyến tính sẽ duyệt toàn bộ không gian tìm kiếm để tìm đối tượng thỏa mãn tiêu chuẩn tìm kiếm **nên các đối tượng này không cần được sắp thứ tự. Nói một cách khác là dữ liệu không cần được tổ chức.**

### 3. HÀM CÀI ĐẶT TRỪU TƯỢNG

- Hàm cài đặt sau tìm một vị trí trong mảng một chiều **a** thỏa điều kiện **T**. Hàm sẽ trả về giá trị -1 nếu không có giá trị nào trong mảng thỏa điều kiện **T**.

- Hàm trừu tượng

```
1. int LinearSearch(KDL a[],  
    int n, điều kiện T)  
2. {  
3.     for(int i=0; i<n; i++)  
4.         if (a[i]thỏa T)  
5.             return i;  
6.     return -1;  
7. }
```

## 4. CẢI TIẾN HÀM CÀI ĐẶT TRỪU TƯỢNG

```
11. int LinearSearch(KDL a[],  
    int n, điều kiện T)  
12. {  
13.     int i=0;  
14.     a[n] = x;  
15.     while(a[i] ko thỏa T)  
16.         i++;  
17.     if(i==n)  
18.         return -1;  
19.     return i;  
20. }
```

## Ứng dụng 1: Viết hàm tìm vị trí giá trị x trong mảng một chiều các số thực.

```
11. int LinearSearch(float a[],  
                      int n, float x)  
12. {  
13.     a[n] = x;  
14.     int i = 0;  
15.     while (a[i] != x)  
16.         i++;  
17.     if (i == n)  
18.         return -1;  
19.     return i;  
20. }
```

## Ứng dụng 1: Viết hàm tìm vị trí giá trị x trong mảng một chiều các số thực.

```
1. int LinearSearch(float a[],  
                    int n, float x)  
2. {  
3.     for(int i=0; i<n; i++)  
4.         if(a[i]==x)  
5.             return i;  
6.     return -1;  
7. }
```



**Ứng dụng 2: Viết hàm tìm vị trí giá trị nhỏ nhất trong mảng một chiều các số thực và phân tích sơ bộ độ phức tạp của thuật toán.**

```
11. int ViTriNhoNhat(float a[],
                    int n)
12. {
13.     int lc=0;
14.     for(int i=0; i<n; i++)
15.         if(a[i]<a[lc])
16.             lc=i;
17.     return lc;
18. }
19. int ViTriNhoNhat(float a[],
                    int n)
20. {
21.     int lc=0;
22.     for(int i=1; i<n; i++)
23.         if(a[i]<a[lc])
24.             lc=i;
25.     return lc;
26. }
```

## 5. ĐỆ QUY VÀ TÌM KIẾM TUYẾN TÍNH

```
1. int LinearSearch(KDL a[], int n,  
    điều kiện T)  
2. {  
3.     if (n==0)  
4.         return -1;  
5.     if (a[n-1]thỏa T)  
6.         return (n-1);  
7.     return LinearSearch  
        (a, n-1, điều kiện T);  
8. }
```

## Ứng dụng 1: Viết hàm tìm vị trí giá trị x trong mảng một chiều các số thực.

```
1. int LinearSearch(float a[],  
                    int n, float x)  
2. {  
3.     if (n==0)  
4.         return -1;  
5.     if (a[n-1]==x)  
6.         return (n-1);  
7.     return LinearSearch(a, n-1, x);  
8. }
```

## 6. BÀI TẬP

- Xét thuật toán tìm tuần tự một số nguyên  $X$  cho trước trong một mảng  $a$  gồm 1000 số nguyên. Trong từng trường hợp sau, hãy cho biết số phần tử tối đa có thể được duyệt trong quá trình tìm kiếm:
  - a. Trường hợp 1: các phần tử của mảng  $a$  chưa được sắp xếp.
  - b. Trường hợp 2: các phần tử của mảng  $a$  được sắp xếp tăng dần.
  - c. Trường hợp 3: các phần tử của mảng  $a$  được sắp xếp giảm dần.

# BÀI TẬP

- a. Trường hợp 1: Số phần tử tối đa có thể được duyệt khi tìm số nguyên  $X$  trong mảng  $a$  khi các phần tử của mảng  $a$  chưa được sắp xếp là: 1.000.
- b. Trường hợp 2: Số phần tử tối đa có thể được duyệt khi tìm số nguyên  $X$  trong mảng  $a$  khi các phần tử của mảng  $a$  được sắp xếp tăng dần là: 1.000.
- c. Trường hợp 3: Số phần tử tối đa có thể được duyệt khi tìm số nguyên  $X$  trong mảng  $a$  khi các phần tử của mảng  $a$  được sắp xếp giảm dần là: 1.000.

# BÀI TẬP

- Cho mảng một chiều các số nguyên  $a$  có 1000 phần tử được sắp tăng dần. Hãy cho biết ta nên áp dụng thuật toán tìm kiếm nào để tìm vị trí giá trị  $x$  có trong mảng và cho biết số phần tử tối đa có thể được duyệt.
  - a. Thuật toán nên áp dụng là thuật toán tìm kiếm nhị phân.
  - b. Số phần tử tối đa có thể được duyệt là  $\log_2 1000 \approx 10$ .