Chương 7: Tìm Kiếm

Ths. Phạm Thanh An
Bộ môn Khoa học máy tính - Khoa CNTT
Trường Đại học Ngân hàng TP.HCM

LOGO

Mục tiêu

- Trình bày các thuật toán thông dụng cho việc tìm kiếm (**Tìm tuần tự**, **tìm nhị phân**)
- Minh họa các thuật toán
- Đánh giá thuật toán

Tầm quan trọng của tìm kiếm

- Tìm kiếm một phần tử trong một tập hợp k đối tượng một thao tác thường sử dụng trong đời sống hằng ngày

Tầm quan trọng của tìm kiếm

- Ví dụ:
  - Cơ sở dữ liệu (Database): tìm 1 sinh viên, tìm 1 tài khoản ngân hàng, tài liệu, quyển sách.
  - Internet: Yahoo!, Google,…

TÌM KIẾM (SEARCHING)

- Định nghĩa
  - Cho n bản ghi $R_1, R_2, \ldots, R_n$, khóa tương ứng k_i
  - Hãy tìm bản ghi có giá trị khóa bằng X
- Kết quả tìm kiếm
  - Thành công: có bản ghi với giá trị khóa X
  - Không thành công: không có bản ghi thích hợp
- Phân loại tìm kiếm
  - Tìm kiếm trong
  - Tìm kiếm ngoài

Tìm kiếm tuần tự (sequential searching)

- Ý tưởng
  - Lần lượt tìm kiếm từ bản ghi đầu tiên cho đến khi tìm thấy, hoặc không còn phần tử để tìm kiếm
  - Thực hiện tìm kiếm trên mảng / danh sách liên kết đơn

Tìm kiếm tuần tự (sequential searching)

- Giải thuật

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

Tìm kiếm tuần tự (sequential searching)

- Độ phức tạp giải thuật
  - Phép toán chính là phép so sánh
  - Trường hợp tốt nhất: 1 phép so sánh
  - Trường hợp xấu nhất: n phép so sánh
  - Trường hợp trung bình: (n+1)/2 phép so sánh

$$\frac{1+2+3+\ldots+n}{n} = \frac{n(n+1)/2}{n} = \frac{(n+1)}{2}$$

Tìm kiếm nhị phân

- Ý tưởng
  - Tiến hành trên dãy đã được sắp xếp tăng dần
  - Dãy tìm kiếm a[1],a[2],…,a[n], khóa tìm kiếm X

1. So sánh khóa X với a[(n+1) div 2]
2. Nếu X=a[(n+1) div 2], kết thúc, ngược lại
   1. Nếu X<a[(n+1) div 2], quay lại (1), tìm kiếm trong dãy a[1],a[2],…,a[(n+1) div 2 - 1]
   2. Nếu X>a[(n+1) div 2], quay lại (1), tìm kiếm trong dãy a[(n+1) div 2 + 1],…,a[n]
3. Kết thúc

Tìm kiếm nhị phân

- Giải thuật

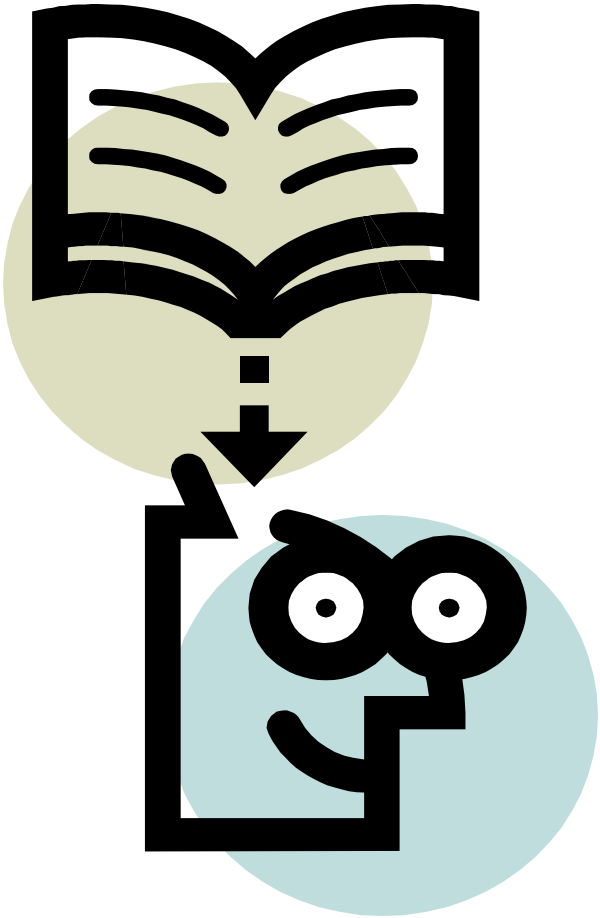
```
int BinarySearch(int a[ ],int l, int r,int x){
        int m
        while (l<=r){
                int m=(l+r)/2;
                if (a[m]==x) return m;
                if (x<a[m]) r=m-1;
                if (x>a[m]) l=m+1;
        }
        return -1;
}
```

Tìm kiếm nhị phân

- Giải thuật

```
 int  BinarySearch1(int a[],int l,int r,int x)
{
        int m=(l+r)/2;
        if (a[m]==x) return m;
        if ( x <a[m]) return BinarySearch1(a,l,m-1,x);
        if (x > a[m]) return BinarySearch1(a,m+1,r,x);
       return -1;



}
```

Q&A