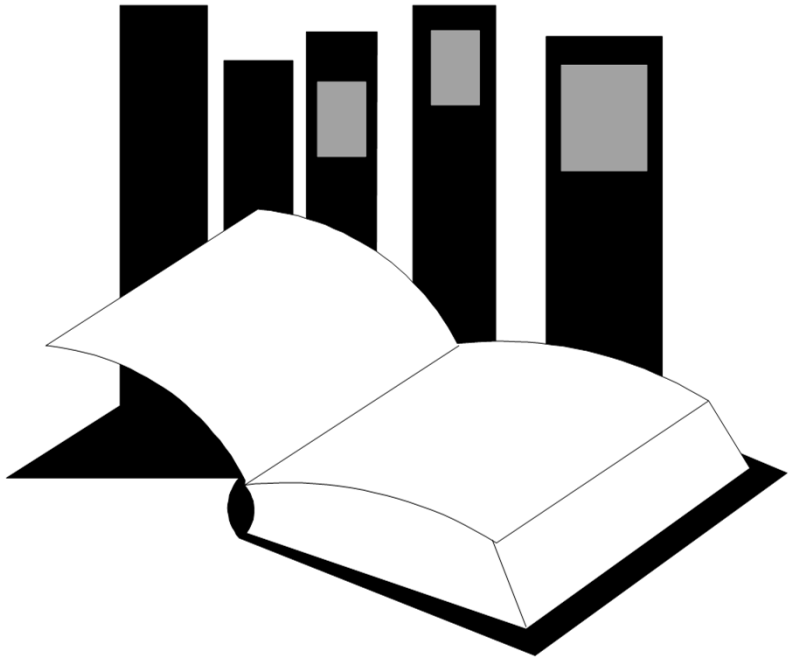
Tìm kiếm tuần tự
Tìm kiếm nhị phân

Nguyễn Tri Tuấn
Khoa CNTT – ĐH.KHTN.Tp.HCM
Email: nttuan@fit.hcmus.edu.vn

Tìm kiếm - Searching



- 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

Công dụng

- Tìm kiếm trong một danh sách các phần tử là một thao tác thường sử dụng trên máy tính
- Ví dụ:
  - Cơ sở dữ liệu (Database): tìm 1 sinh viên, tìm 1 tài khoản ngân hàng,…
  - Internet: Yahoo!, Google,…

Các phương pháp phổ biến

- Tìm tuần tự (Serial Search)
  - Đơn giản
  - Chi phí **O(n)**
- Tìm nhị phân
  - Phải là 1 **danh sách "đặc"**
  - Dữ liệu cần được sắp thứ tự
  - Chi phí trung bình **$O(\log_2 n)$**

Tìm tuần tự
(Serial Search)

```
int  SerialSearch(int a[], int n, int key)
{
   for (int i=0; i < n; i++)
        if (a[i]==key)     return i;      // tìm thấy
   return –1;         // không tìm thấy
}
```

Serial Search
Đánh giá thuật toán

- Kích thước của dãy: n
- Trường hợp tốt nhất: O(1), key==a[0]
- Trường hợp xấu nhất: O(n), key==a[n-1] hoặc không tìm thấy
- Trường hợp trung bình:
  - ít hơn O(n)
  - Chính xác là bao nhiêu ?

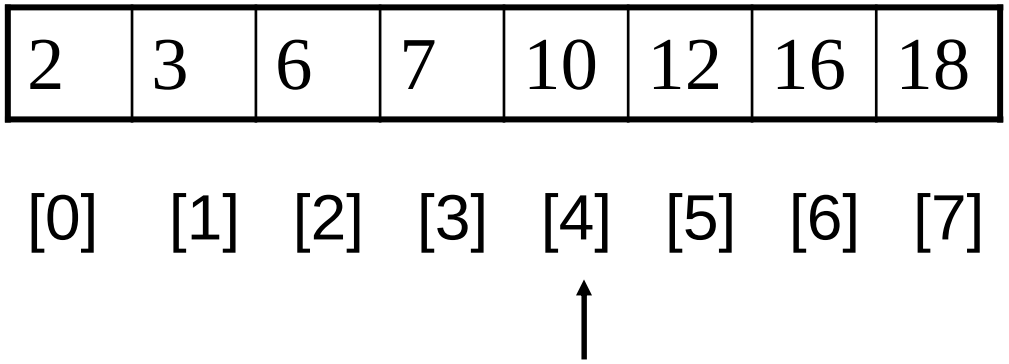
Serial Search
Trường hợp trung bình

- Giả sử:
  - phần tử cần tìm *key* có trong dãy
  - xác suất xuất hiện tại các vị trí trong dãy là như nhau
- Chi phí trung bình:

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

Tìm nhị phân (Binary Search)

- Các phần tử được sắp
  - n = 8
  - key = 16



- Xét phần tử giữa

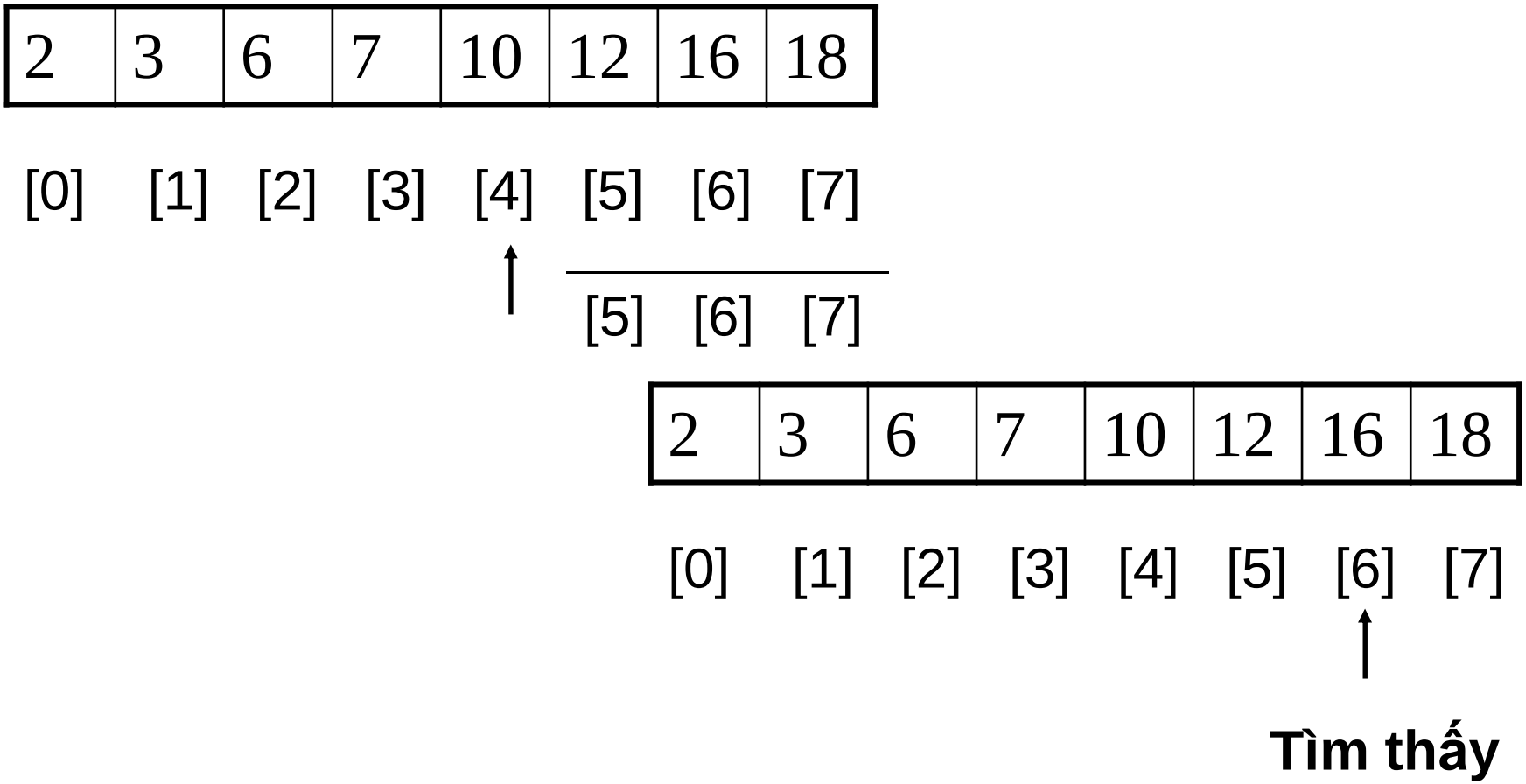
  m = n/2
- Nếu (a[m]==key)

  → Kết thúc !
- Nếu (key < a[m])

  Xét ½ dãy bên trái
- Nếu (key > a[m])

  Xét ½ dãy bên phải

Tìm nhị phân (Binary Search)



Binary Search
(Minh họa chương trình)

```
int  BinarySearch(int a[], int n, int key)
{
    int Left = 0, Right = n-1;
    while (Left <= Right) {
        int Mid = (Left + Right)/2;
        if (a[Mid]==key)        return Mid;      // tìm thấy
        else if (key < a[Mid])  Right = Mid – 1;
            else Left = Mid + 1;
    }
    return –1;    // không tìm thấy
}
```