

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI – BÀI GIẢNG ĐIỆN TỬ

BÀI 3

CÁC PP TÌM KIẾM VÀ SẮP XẾP

3.1. CÁC PHƯƠNG PHÁP TÌM KIẾM

© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Nội dung

1. Khái niệm tìm kiếm
2. Phát biểu bài toán tìm kiếm
3. Các phương pháp tìm kiếm
 - Tìm kiếm trên dãy chưa sắp xếp.
 - Tìm kiếm trên dãy đã sắp xếp.

© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

3.1.1. Khái niệm tìm kiếm

• Đặt vấn đề



© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Khái niệm tìm kiếm (tt)

- **Khái niệm**
 - Tìm kiếm là việc kiểm tra xem có hay không một đối tượng có một số thông tin cho trước (**đối tượng cần tìm**) trong một tập các đối tượng cho trước (**không gian tìm kiếm**).
- **Ví dụ: Tìm một chùm chìa khóa trong một gian phòng.**
 - Ta có hình ảnh của chùm chìa khóa.
 - Gian phòng gồm nhiều đồ đạc.

© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

3.1.2. Phát biểu bài toán tìm kiếm

Đầu vào: - Dãy X , có n đối tượng, mỗi đối tượng có một "khóa tìm kiếm".
- Khóa của đối tượng cần tìm (**Key**).

Đầu ra: - Vị trí của đối tượng có khóa 'Key' trong dãy X nếu tìm kiếm thành công, ngược lại trả về -1.

Ví dụ:

X0	X1	X2	X3	X4
5	1	6	8	2

Số $k = 6$

Đầu ra: $v = 2$ (vị trí của số k trong dãy X)

© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

3.1.3. Các thuật toán tìm kiếm

- Tùy theo dữ liệu vào ta có thể phân chia bài toán tìm kiếm thành hai loại.
 - **Tìm kiếm trên dãy chưa sắp:**
 - ✓ Dãy tìm kiếm chưa được sắp xếp theo thứ tự khóa tìm kiếm.
 - **Tìm kiếm trên dãy đã sắp:**
 - ✓ Dãy tìm kiếm đã sắp theo thứ tự của khóa tìm kiếm.

© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

3.1.4. Tìm kiếm trên dãy chưa sắp xếp

- Với một dãy chưa được sắp xếp thì cách tìm kiếm duy nhất là tìm kiếm tuần tự.
- Tìm kiếm tuần tự là một phương pháp tìm kiếm khá phổ biến và hết sức đơn giản.

© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

3.1.4.1. Tìm kiếm tuần tự

- Không gian tổng quát, ta tìm hiểu phương pháp tìm kiếm tuần tự với việc tìm kiếm số nguyên k trên dãy $x[]$ có n số nguyên.
- Phương pháp tìm kiếm:
 - Duyệt dãy $x[]$, so sánh số k lần lượt với các số $x[i]$ trong dãy ($i = 0, 1, \dots, n-1$).
 - Nếu $k = x[i]$ thì: kết thúc tìm kiếm, trả về i (thành công).
 - Ngược lại thì ($k \neq x[i]$):
 - Nếu chưa duyệt hết dãy $x[]$ ($i < n$) chuyển sang số tiếp theo.
 - Ngược lại, kết thúc tìm kiếm, trả về -1 (không thành công).

© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự (tt)

- Ví dụ: Cho dãy số nguyên $x[]$ và số nguyên $k = 6$.

x_0	x_1	x_2	x_3	x_4
5	1	6	8	2

$k = 6$

- Yêu cầu: Minh họa việc tìm kiếm số k trong dãy x .

© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự (tt)

Việc tìm kiếm cũng có thể minh họa như sau

- $\rightarrow i = 0 < n$; $x_0 = 5 \neq k = 6$; $i = i + 1$; Chuyển sang đối tượng kế tiếp
- $\rightarrow i = 1 < n$; $x_1 = 1 \neq k = 6$; $i = i + 1$; Chuyển sang đối tượng kế tiếp
- $\rightarrow i = 2 < n$; $x_2 = 6 = k = 6$; Kết thúc tìm kiếm

Tìm kiếm thành công với vị trí tìm được: $v = i = 2$

© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự (tt)

- Ví dụ 2: Cho dãy số nguyên x và số hai số nguyên $k_1 = 42$ và $k_2 = 43$.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
3	48	11	36	25	23	42	7

- Minh họa việc tìm số $k_1 = 42$ và số $k_2 = 43$ trong dãy bằng phương pháp tìm kiếm tuần tự.

© 2021 Hanoi University of Industry All rights reserved

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự (tt)

- Giải thuật

```

graph TD
    START([START]) --> I0[i = 0]
    I0 --> D1{<math>(i < n) \ \&\& \ (key \neq x[i], key)</math>}
    D1 -- Yes --> Iplus[i = i + 1]
    Iplus --> D1
    D1 -- No --> D2{<math>(i < n)</math>}
    D2 -- Yes --> Vset[v = i]
    D2 -- No --> Vsetminus1[v = -1]
    Vset --> STOP([STOP])
    Vset\1 --> STOP
  
```

© 2021 Hanoi University of Industry All rights reserved

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự (tt) - Ứng dụng

- Viết chương trình thực hiện các việc sau
 - Nhập vào một dãy X, có n số nguyên ($0 < n < 100$, n nhập từ bàn phím).
 - Nhập số nguyên k.
 - Bảng phương pháp tìm kiếm tuần tự hãy cho biết số k có xuất hiện trong dãy không, nếu có thì nó ở vị trí thứ bao nhiêu.

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự cải tiến (tt)

- Nhận thấy mỗi lần so sánh đều phải kiểm tra xem dãy đã hết chưa ($i < n$), nên tốn thêm khá nhiều thời gian.
- Để tránh điều đó người ta thêm đối tượng có khóa key vào cuối dãy x ($x[n].key = key \rightarrow$ cải tiến).

Ví dụ

x0	x1	x2	x3	x4
5	1	6	8	2

x0	x1	x2	x3	x4	x5
5	1	6	8	2	6

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự cải tiến (tt)

- Giải thuật

```

graph TD
    START([START]) --> Init["i = 0; x[n].key = key;"]
    Init --> Decision1{"key != x[i].key"}
    Decision1 -- Yes --> IncI["i = i + 1"]
    IncI --> Decision1
    Decision1 -- No --> Decision2{"i < n"}
    Decision2 -- Yes --> DecV["v = i;"]
    DecV --> DecV1["v = -1;"]
    DecV1 --> STOP([STOP])
    Decision2 -- No --> STOP
  
```

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự cải tiến - Ứng dụng

- Viết chương trình thực hiện các việc sau
 - Nhập vào một dãy X, có n số nguyên ($0 < n < 100$, n nhập từ bàn phím).
 - Nhập số nguyên k.
 - Bảng phương pháp tìm kiếm tuần tự cải tiến hãy cho biết số k có xuất hiện trong dãy không, nếu có thì nó ở vị trí thứ bao nhiêu.

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Bài tập

- Viết chương trình thực hiện các việc sau
 - Nhập vào một danh sách học sinh ($0 < n < 100$, n nhập từ bàn phím), mỗi học sinh gồm các thông tin: Mã học sinh, họ và tên, năm sinh và điểm trung bình.
 - In danh sách vừa nhập ra màn hình.
 - Nhập vào họ tên của một học sinh, bằng phương pháp tìm kiếm tuần tự (cải tiến) cho biết học sinh có họ tên vừa nhập có trong danh sách trên hay không, nếu có hãy in thông tin đầy đủ của học sinh này ra màn hình.

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

3.1.5. Tìm kiếm trên dãy đã sắp xếp

- Với một dãy đã sắp xếp theo thứ tự của khóa tìm kiếm, thì việc tìm kiếm, về cơ bản sẽ nhanh hơn.
- Việc tìm kiếm có thể thực hiện bằng một trong hai phương pháp.
 - Tìm kiếm tuần tự.
 - Tìm kiếm nhị phân.

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

3.1.5.1. Tìm kiếm tuần tự

- Việc tìm kiếm về cơ bản giống như tìm kiếm trên dãy chưa sắp.
- Quá trình tìm kiếm kết thúc khi gặp một trong 3 điều kiện.
 - Gặp đối tượng có khóa bằng với khóa của đối tượng cần tìm (tìm kiếm thành công).
 - Gặp đối tượng có khóa "lớn hơn" khóa của đối tượng cần tìm (tìm kiếm không thành).
 - Đã duyệt hết dãy (tìm kiếm không thành).

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự (tt)

- Ví dụ: Cho dãy số nguyên $x[]$ được sắp xếp theo chiều tăng dần và số nguyên $k = 4$.

x_0	x_1	x_2	x_3	x_4
1	2	5	6	8

$k = 4$
- Yêu cầu: Minh họa việc tìm kiếm số k trong dãy x .

Kết quả tìm kiếm
 $v = -1$

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự (tt)

- Giải thuật**

```

graph TD
    START([START]) --> I0[i = 0]
    I0 --> Cond1{<math>i < n \ \&\& \ (key > x[i].key)</math>}
    Cond1 -- Yes --> Iplus1[i = i + 1]
    Iplus1 --> Cond1
    Cond1 -- No --> Cond2{<math>i < n \ \&\& \ (x[i].key = key)</math>}
    Cond2 -- Yes --> Vplus1[v = i]
    Vplus1 --> STOP([STOP])
    Cond2 -- No --> Vminus1[v = -1]
    Vminus1 --> STOP
  
```

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm tuần tự (tt) - Ứng dụng

- Viết chương trình thực hiện các yêu cầu:
 - Nhập vào một dãy X , có n số nguyên ($0 < n < 100$, n nhập từ bàn phím).
 - Sắp xếp dãy theo chiều tăng dần (bằng một trong các thuật toán sắp xếp đã học).
 - Nhập số nguyên k .
 - Bảng phương pháp tìm kiếm tuần tự (trên dãy đã sắp) hãy cho biết số k có xuất hiện trong dãy không, nếu có thì nó ở vị trí thứ bao nhiêu.

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

3.1.5.2. Tìm kiếm nhị phân

- Với một dãy được sắp xếp theo thứ tự của khóa tìm kiếm thì phương pháp tìm kiếm nhị phân sẽ "tốt hơn".
- Việc tìm kiếm được thực hiện như sau:
 - Với một dãy khác rỗng, so sánh khóa cần tìm với khóa của đối tượng ở trung tâm của dãy đang xét (Giả sử dãy đang xét là x_0, x_{i+1}, \dots, x_i , thì phần tử trung tâm là x_m với $m = (i + 0) / 2$).
 - Tìm kiếm kết thúc thành công nếu $x[m].key = key$
 - Nếu $x[m].key > key$, thực hiện tìm kiếm với dãy trái x_0, \dots, x_{m-1}
 - Nếu $x[m].key < key$, thực hiện tìm kiếm với dãy phải x_{m+1}, \dots, x_i
 - Với dãy khóa kế tiếp, một kỹ thuật tương tự lại được sử dụng.
 - Quá trình tìm kiếm tiếp tục cho đến khi gặp đối tượng cần tìm (thành công) hoặc dãy khóa đang xét trở nên rỗng (không thành công).

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tìm kiếm nhị phân (tt)

- Ví dụ: Cho dãy số nguyên $x[]$ được sắp xếp theo chiều tăng dần và số nguyên $k = 4$.

x_0	x_1	x_2	x_3	x_4
1	2	5	6	8

$k = 4$
- Yêu cầu: Minh họa việc tìm kiếm số k trong dãy x .

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tim kiếm nhị phân (tt)

- Mình họa việc tìm kiếm số k trong dãy $x[]$ bằng phương pháp tìm kiếm nhị phân.

$k = 4$, $X_m = 5$, $l = 0$, $r = 4$, $m = 2$
 $k < X_m$
 $k = 4$, $X_m = 1$, $l = 0$, $r = 1$, $m = 0$
 $k > X_m$
 $k = 4$, $X_m = 2$, $l = 1$, $r = 1$, $m = 1$
 $k > X_m$
 $k = 4$, $X_m = 5$, $l = 2$, $r = 1$

Trường hợp dãy đang xét trở nên rỗng ($l > r$), vì thế tìm kiếm kết thúc không thành công ($v = -1$)

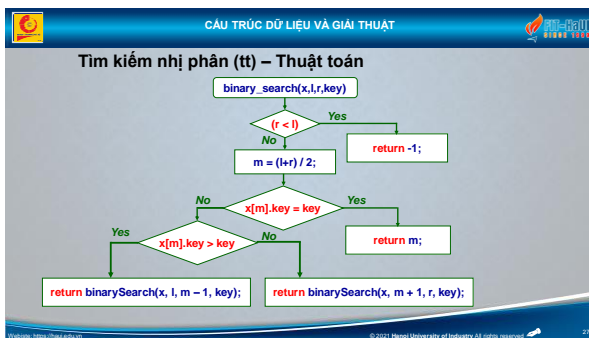
CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tim kiếm nhị phân (tt)

- Ví dụ 2:
 - Cho dãy số $x[]$ được sắp xếp theo chiều tăng dần

$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$	$x[5]$	$x[6]$	$x[7]$
3	7	11	23	25	36	42	48

- Yêu cầu:
 - Mình họa việc tìm số $k_1 = 11$ và số $k_2 = 37$ trong dãy bằng phương pháp tìm kiếm nhị phân.



CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tim kiếm nhị phân (tt)

- Hàm tìm kiếm nhị phân viết theo phương pháp đệ quy

```

int binarySearch(int x[], int l, int r, int key)
{
    if (r < l) return -1; //Dãy rỗng
    else {
        int m = (l+r) / 2;
        if (x[m] == key) return m;
        else if (x[m] > key)
            return binarySearch(x, l, m - 1, key);
        else return binarySearch(x, m + 1, r, key);
    }
}
  
```

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT


Tim kiếm nhị phân (tt) - Ứng dụng

- Viết chương trình thực hiện các yêu cầu:
 - Nhập vào một dãy X , có n số nguyên ($0 < n < 100$, n nhập từ bàn phím).
 - Sắp xếp dãy theo chiều tăng dần (bằng một trong các thuật toán sắp xếp đã học).
 - Nhập số nguyên k .
 - Bằng phương pháp tìm kiếm nhị phân hãy cho biết số k có xuất hiện trong dãy không, nếu có thì nó ở vị trí thứ bao nhiêu.


CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Tim kiếm nhị phân (tt) - Ứng dụng

- Viết chương trình thực hiện các việc sau
 - Nhập vào một danh sách học sinh ($0 < n < 100$, n nhập từ bàn phím), mỗi học sinh gồm các thông tin: Mã học sinh, họ và tên, năm sinh và điểm trung bình.
 - Sắp xếp danh sách theo chiều giảm dần của điểm trung bình.
 - Nhập vào điểm trung bình của một học sinh, bằng phương pháp tìm kiếm nhị phân cho biết học sinh có điểm trung bình vừa nhập có trong danh sách trên hay không, nếu có hãy in thông tin đầy đủ của học sinh này ra màn hình.



CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT



- **Bài tập 1:** Cho dãy số nguyên
34 14 24 54 84 64 94 74 04
 - Minh họa việc tìm kiếm số $k_1 = 94$, số $k_2 = 45$ trong dãy trên bằng phương pháp tìm kiếm tuần tự.
 - Cài đặt chương trình tìm kiếm số k_1 , k_2 trong dãy số trên bằng phương pháp đã minh họa.
- **Bài tập 2:** Cho dãy được sắp xếp theo chiều tăng dần:
04 14 24 34 54 64 74 84 94
 - Minh họa việc tìm kiếm số $k_1 = 64$, số $k_2 = 44$ trong dãy bằng phương pháp tìm kiếm nhị phân.
 - Cài đặt chương trình tìm kiếm số k_1 , k_2 trong dãy bằng phương pháp tìm kiếm đã minh họa.

[Website: https://hust.edu.vn](https://hust.edu.vn)

© 2021 Hanoi University of Technology. All rights reserved.

21