

CHƯƠNG 6

CÁC THUẬT TOÁN TÌM KIẾM

Design by Minh An

Email: minhav78@gmail.com

Nội dung

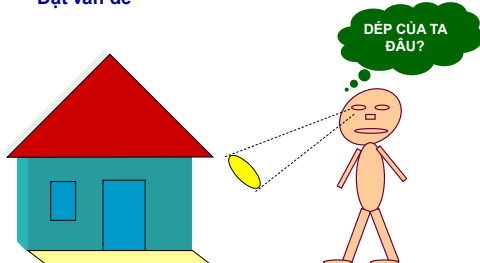
1. Khái niệm tìm kiếm
2. Phát biểu bài toán tìm kiếm
3. Các phương pháp tìm kiếm
 - Tìm kiếm trên dãy chưa sắp xếp.
 - Tìm kiếm trên dãy đã sắp xếp.

Design by Minh An

Email: minhav78@gmail.com

6.1. Khái niệm tìm kiếm

- Đặt vấn đề



Design by Minh An

Email: minhav78@gmail.com

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.

Design by Minh An

Email: minhav78@gmail.com

6.2. 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:

Đầu vào:	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)				

Design by Minh An

Email: minhav78@gmail.com

6.3. Các thuật toán tìm kiếm

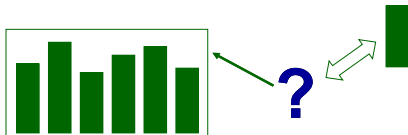
- Tùy theo dữ liệu vào ta có thể phân chia bài toán tìm kiếm thành hai loại.
 - Tìm kiếm trên dãy chưa sắp:
 - ✓ Dãy tìm kiếm chưa được sắp xếp theo thứ tự khóa tìm kiếm.
 - Tìm kiếm trên dãy đã sắp:
 - ✓ Dãy tìm kiếm đã sắp theo thứ tự của khóa tìm kiếm.

Design by Minh An

Email: minhav78@gmail.com

6.3.1. 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.



Design by Minh An

Email: minhav78@gmail.com

6.3.1.1. Tìm kiếm tuần tự

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

Design by Minh An

Email: minhav78@gmail.com

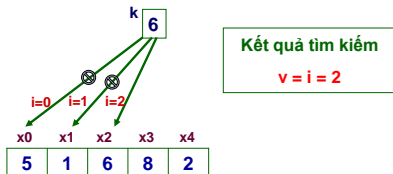
Tìm kiếm tuần tự (tt)

- 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 .



Design by Minh An

Email: minhav78@gmail.com

Tìm kiếm tuần tự (tt)

Việc tìm kiếm cũng có thể minh họa như sau

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

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

Design by Minh An

Email: minhav78@gmail.com

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

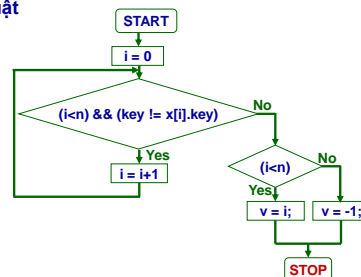
- Minh họa việc tìm số $k_1 = 42$ và số $k_2 = 43$ trong dãy bằng phương pháp tìm kiếm tuần tự.

Design by Minh An

Email: minhav78@gmail.com

Tìm kiếm tuần tự (tt)

- Giải thuật



Design by Minh An

Email: minhav78@gmail.com

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.
 - Bằng 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.

Design by Minh An

Email: minhav78@gmail.com

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 ($x[n].key = \text{key} \rightarrow$ cải tiến).

Ví dụ

x0	x1	x2	x3	x4
5	1	6	8	2

→

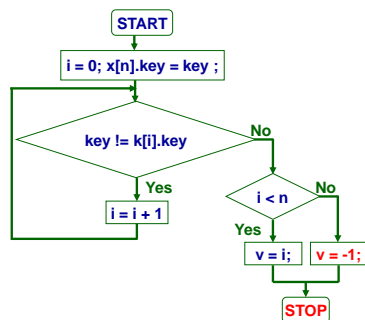
x0	x1	x2	x3	x4	x5
5	1	6	8	2	6

Design by Minh An

Email: minhav78@gmail.com

Tìm kiếm tuần tự cải tiến (tt)

Giải thuật



Design by Minh An

Email: minhav78@gmail.com

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.
 - Bằng 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.

Design by Minh An

Email: minhav78@gmail.com

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.

Design by Minh An

Email: minhav78@gmail.com

6.3.2. 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.

Design by Minh An

Email: minhav78@gmail.com

6.3.2.1. 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).

Design by Minh An

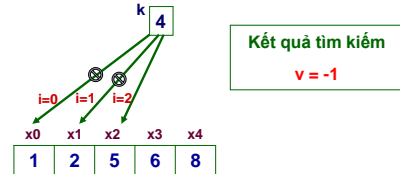
Email: minhav78@gmail.com

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	6	$k = 4$

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

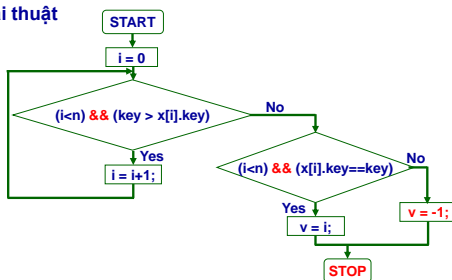


Design by Minh An

Email: minhav78@gmail.com

Tìm kiếm tuần tự (tt)

Giải thuật



Design by Minh An

Email: minhav78@gmail.com

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 .
- Bằng 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.

Design by Minh An

Email: minhav78@gmail.com

6.3.2.2. Tìm kiếm nhị phân

- Với một dãy được sắp theo thứ tự khóa thì phương pháp tìm kiếm nhị phân sẽ "tốt hơn".
- Việc tìm kiếm được thực hiện như sau:
 - So sánh khóa cần tìm với khóa của đối tượng ở trung tâm của dãy đang xét (Giả sử dãy đang xét là x_i, x_{i+1}, \dots, x_j thì phần tử trung tâm là x_m với $m = (i+j) \text{ div } 2$.)
 - Tìm kiếm kết thúc thành công nếu $x[m].key = key$
 - Nếu $x[m].key > key$, thực hiện tìm kiếm với dãy trái x_i, \dots, x_{m-1}
 - Nếu $x[m].key < key$, thực hiện tìm kiếm với dãy phải x_{m+1}, \dots, x_j
 - 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 mong muốn (thành công) hoặc dãy khóa đang xét trở THÀNH RỖNG.

Design by Minh An

Email: minhav78@gmail.com

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 .

Design by Minh An

Email: minhav78@gmail.com

Tìm kiếm nhị phân (tt)

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



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

Design by Minh An

Email: minhav78@gmail.com

Tìm 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

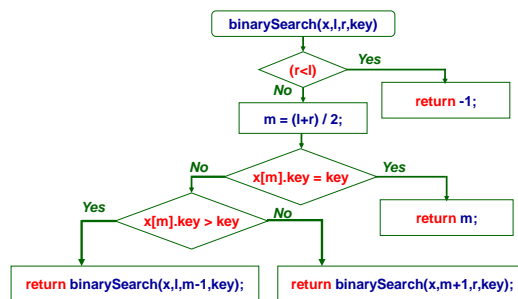
- Yêu cầu:

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

Design by Minh An

Email: minhav78@gmail.com

Tìm kiếm nhị phân (tt) – Thuật toán



Design by Minh An

Email: minhav78@gmail.com

Tìm kiếm nhị phân (tt)

- Hàm tìm kiếm nhị phân

```

int binarySearch(int x[],int l,int r,int key)
{
    if (r < l) return -1;
    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);
    }
}
  
```

Design by Minh An

Email: minhav78@gmail.com

Tìm 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.

Design by Minh An

Email: minhav78@gmail.com

Tìm 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.

Design by Minh An

Email: minhav78@gmail.com