

Cấu trúc dữ liệu và giải thuật

CÁC CHIẾN LƯỢC TÌM KIẾM

Giảng viên:

Văn Chí Nam – Nguyễn Thị Hồng Nhung – Đặng Nguyễn Đức Tiến

Nội dung trình bày

2

Giới thiệu

Tìm kiếm tuần tự

Tìm kiếm nhị phân

Tìm kiếm theo bảng băm

Tổng kết

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Giới thiệu

3

- ◉ Thao tác tìm kiếm rất phổ biến trong cuộc sống hàng ngày.
 - ▣ Tìm kiếm hồ sơ, tập tin.
 - ▣ Tìm kiếm tên người trong danh sách.
 - ▣ ...

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm

4

- ◉ Có nhiều loại:
 - ▣ Tìm kiếm tuần tự (Sequential/ Linear Search)
 - ▣ Tìm kiếm nhị phân (Binary Search)
 - ▣ ...
- ◉ Mục tiêu:
 - ▣ Tìm hiểu về 2 thuật toán tìm kiếm cơ bản.
 - ▣ Phân tích thuật toán để lựa chọn thuật toán phù hợp khi áp dụng vào thực tế.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

5

Tìm kiếm tuần tự

Sequential Search

Linear Search

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm tuần tự

6

- ◉ **Input:**
 - ▣ Dãy A , n phần tử
 - ▣ Giá trị x cần tìm
- ◉ **Output:**
 - ▣ Nếu x xuất hiện trong A : trả về vị trí xuất hiện đầu tiên của x
 - ▣ Nếu không: trả về n hoặc -1
- ◉ **Thuật toán:**
 - ▣ Vét cạn (exhaustive)
 - ▣ Dùng lính canh (sentinel)

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Tìm kiếm tuần tự - Vết cạn

7

Thuật toán:

▣ Lần lượt so sánh x với các phần tử của mảng A cho đến khi gặp được phần tử cần tìm, hoặc hết mảng.

▣ Ví dụ: $A = \{1, 25, 6, 5, 2, 37, 40\}$, $x = 6$

$x = 6$
↓

1	25	6	5	2	37	40
---	----	---	---	---	----	----

$x = 6$
↓

1	25	6	5	2	37	40
---	----	---	---	---	----	----

$x = 6$
↓

1	25	6	5	2	37	40
---	----	---	---	---	----	----

➡ **Dừng**

Tìm kiếm tuần tự - Vết cạn

8

Thuật toán: LinearExhaustive

- **Bước 1.** Khởi tạo biến chỉ số: $i = 0$
- **Bước 2.** Kiểm tra xem có thực hiện hết mảng hay chưa: **So sánh i và n**
 - Nếu chưa hết mảng ($i < n$), sang bước 3.
 - Nếu đã hết mảng ($i \geq n$), thông báo không tìm thấy giá trị x cần tìm.
- **Bước 3.** **So sánh giá trị $a[i]$ với giá trị x cần tìm**
 - Nếu $a[i]$ bằng x : Kết thúc chương trình và thông báo đã tìm thấy x .
 - Nếu $a[i]$ khác x , **tăng i thêm 1** và quay lại bước 2.

Cấu trúc dữ liệu và giải thuật - HCMUS 2011

Tìm kiếm tuần tự - Vết cạn

9

- Nhận xét: Phép so sánh là phép toán sơ cấp được dùng trong thuật toán. Suy ra, số lượng các phép so sánh sẽ là thước đo độ phức tạp của thuật toán.
- Mỗi vòng lặp có 2 điều kiện cần kiểm tra:
 - ▣ Kiểm tra cuối mảng (bước 2)
 - ▣ Kiểm tra phần tử hiện tại có bằng x ? (bước 3)

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Tìm kiếm tuần tự - Vết cạn

10

- Trường hợp x nằm ở 2 biên của mảng A : rất hiếm khi xuất hiện.
- Ước lượng số vòng lặp trung bình sẽ hữu ích hơn.
- Số phép so sánh trung bình:
$$2(1+2+ \dots + n)/n = n+1$$
$$\Rightarrow \text{Số phép so sánh tăng/giảm tuyến tính theo số phần tử}$$

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Tìm kiếm tuần tự - Vết cạn

11

- Vậy độ phức tạp của thuật toán là:
 - ▣ Tốt nhất: $O(1)$.
 - ▣ Trung bình: $O(n)$.
 - ▣ Xấu nhất: $O(n)$.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Tìm kiếm tuần tự - Lính canh

12

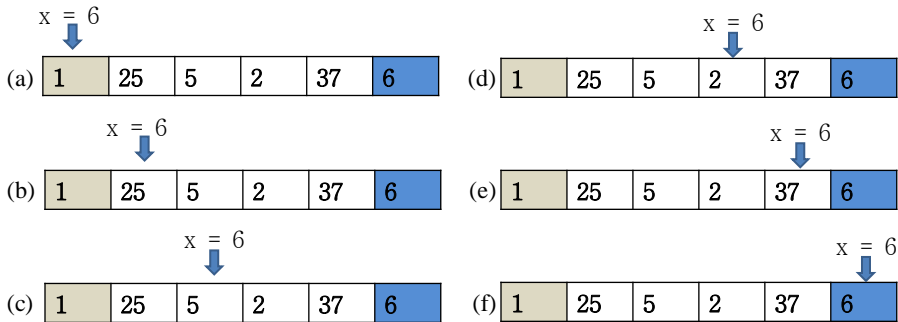
- Trong thuật toán vết cạn, có 2 điều kiện được kiểm tra.
- Có thể bỏ việc kiểm tra điều kiện cuối mảng bằng cách dùng “lính canh”.
- Lính canh là phần tử có giá trị bằng với phần tử cần tìm và đặt ở cuối mảng.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Tìm kiếm tuần tự - Lính canh

13

◉ Ví dụ: $A = \{1, 25, 5, 2, 37\}$, $x = 6$



➡ **return 5;**

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Tìm kiếm tuần tự - Lính canh

14

Thuật toán: **LinearSentinel**

- **Bước 1.** Khởi tạo biến chỉ số: $i = 0$
- **Bước 2.** So sánh giá trị $a[i]$ với giá trị x cần tìm
 - Nếu $a[i]$ bằng x :
 - Nếu $i < n$: Kết thúc chương trình và thông báo đã tìm thấy x .
 - Nếu $i \geq n$: Thông báo không tìm thấy x trong mảng.
 - Nếu $a[i]$ khác x , **tăng i thêm 1** và quay lại bước 2.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Tìm kiếm tuần tự - Lính canh

15

- ◉ Thực nghiệm cho thấy trong trường hợp n lớn, thời gian tìm kiếm giảm khi dùng phương pháp lính canh.
 - ▣ Với $n=15000$: nhanh hơn khoảng 20% (0,22s so với 0,28s)

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

16

Tìm kiếm nhị phân

Binary Search

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

17

- Với dãy A được sắp xếp thứ tự (ví dụ: tăng dần), độ phức tạp của thuật toán tìm kiếm tuần tự không đổi.
- Tận dụng thông tin của mảng đã được sắp xếp để giới hạn vị trí của giá trị cần tìm trong mảng.
-> Thuật toán tìm kiếm nhị phân.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

18

- **Input:**
 - ▣ Dãy A , n phần tử **đã được sắp xếp**
 - ▣ Giá trị x cần tìm
- **Output:**
 - ▣ Nếu x xuất hiện trong A : trả về một vị trí xuất hiện của x
 - ▣ Nếu không: trả về n hoặc -1

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

19

◉ Ý tưởng:

- ▣ So sánh x với phần tử chính giữa mảng A .
 - Nếu x là phần tử giữa thì dừng.
- ▣ Nếu không: xác định xem x có thể thuộc nửa trái hay nửa phải của A .
- ▣ Lặp lại 2 bước trên với nửa đã được xác định.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

20

Thuật toán: BinarySearch($A[], n, x$)

- ◉ **Bước 1.** Khởi gán $left = 0$ và $right = n - 1$.
- ◉ **Bước 2.** Trong khi $left \leq right$, thực hiện:
 - ▣ 2.1. Đặt $mid = (left + right)/2$
 - ▣ 2.2. So sánh giá trị x và $a[mid]$:
 - Nếu $x < a[mid]$, gán $right = mid - 1$.
 - Nếu $x > a[mid]$, gán $left = mid + 1$.
 - Nếu $x = a[mid]$, thông báo đã tìm thấy x và kết thúc.
- ◉ Kết quả trả về không tìm thấy x nếu $left > right^*$.

** Điều này có nghĩa là không còn phần tử nào trong mảng: x không có trong mảng*

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

21

Cài đặt đệ quy: BinarySearch(A[], left, right, x)

◉ **Bước 1.** Nếu $\text{left} > \text{right}$: thông báo không tìm thấy x và thoát khỏi hàm.

◉ **Bước 2.**

▣ 2.1. Đặt $\text{mid} = (\text{left} + \text{right})/2$

▣ 2.2. So sánh giá trị x và $a[\text{mid}]$:

■ Nếu $x < a[\text{mid}]$, Gọi BinarySearch(A, left, mid - 1, x)

■ Nếu $x > a[\text{mid}]$, Gọi BinarySearch(A, mid + 1, right, x)

■ Nếu $x = a[\text{mid}]$, thông báo đã tìm thấy x và kết thúc (trả lại giá trị mid)

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

22

◉ **Minh họa:**

▣ $A[] = \{1, 2, 6, 26, 28, 37, 40\}$, $x = 2$

index	0	1	2	3	4	5	6
A[i]	1	2	6	26	28	37	40
Vòng 1	left			mid			right
Vòng 2	left	mid	right				



$x = a[1] \rightarrow \text{return } 1$

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

23

◉ Minh họa:

▣ $A[] = \{1, 2, 6, 26, 28, 37, 40\}$, $x = 40$

index	0	1	2	3	4	5	6
A[i]	1	2	6	26	28	37	40
Vòng 1	left			mid			right
Vòng 2					left	mid	right
Vòng 3							left mid right



$x = a[6] \rightarrow \text{return } 6$

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

24

◉ Minh họa:

▣ $A[] = \{1, 2, 6, 26, 28, 37, 40\}$, $x = -7$

index	0	1	2	3	4	5	6
A[i]	1	2	6	26	28	37	40
Vòng 1	left			mid			right
Vòng 2	left	mid	right				
Vòng 3	left mid right						
Vòng 4							

$\text{right} = -1, \text{left} = 0$
 $\Rightarrow \text{right} < \text{left} \Rightarrow$ thoát khỏi while,
return -1

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

25

◉ Phân tích thuật toán tuyến tính:

- ▣ Mỗi lần lặp thì chiều dài của mảng con giảm *khoảng* $\frac{1}{2}$ so với mảng trước đó.
- ▣ $n = 2^k + m$ ($0 \leq m < 2$)
- ▣ $2^k \leq n < 2^{k+1} \Rightarrow k \leq \log_2 n < k+1 \Rightarrow k = \lfloor \log_2 n \rfloor$
 \Rightarrow mảng A ban đầu được chia nửa *khoảng* **k** lần.
- ▣ Số lần thực hiện vòng while là khoảng k lần, mỗi vòng lặp thực hiện 1 phép so sánh.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

26

◉ Phân tích thuật toán tuyến tính:

- ▣ Trường hợp tốt nhất: $k = 1 \Leftrightarrow x$ là phần tử chính giữa của mảng.
- ▣ Trường hợp xấu nhất: $k = \lfloor \log_2 n \rfloor + 1 \Leftrightarrow x$ không thuộc mảng hoặc x là phần tử cuối cùng của mảng
 \Rightarrow Số phép so sánh tăng theo hàm logarit

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán tìm kiếm nhị phân

27

- ◉ Độ phức tạp của tìm kiếm nhị phân
 - ▣ Trường hợp tốt nhất: $O(1)$
 - ▣ Trường hợp trung bình: $O(\log_2 n)$
 - ▣ Trường hợp xấu nhất: $O(\log_2 n)$

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

So sánh hiệu suất

28

- ◉ So sánh trường hợp xấu nhất của 2 thuật toán:

Kích thước mảng	T/h xấu nhất	
	Tuần tự	Nhị phân
100.000	100.000	16
200.000	200.000	17
400.000	400.000	18
800.000	800.000	19
1.600.000	1.600.000	20

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Tổng kết

29

- Có nhiều thuật toán tìm kiếm, ước lượng số phép so sánh của mỗi thuật toán cho biết hiệu suất của thuật toán.
- Thuật toán tuần tự tìm kiếm cho đến khi tìm thấy giá trị cần tìm hoặc hết mảng
- Hiệu suất của tìm kiếm tuần tự trong trường hợp xấu nhất là 1 hàm tuyến tính theo số phần tử mảng.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Tổng kết

30

- Nếu mảng đã được sắp xếp thì nên dùng tìm kiếm nhị phân.
- Tìm kiếm nhị phân dùng kết quả của phép so sánh để thu hẹp vùng tìm kiếm kế tiếp.
- Hiệu suất của tìm kiếm nhị phân là một hàm logarit theo số phần tử mảng.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011