



C Programming Basic – week 7

Nội dung

Tìm kiếm nhị phân

Tìm kiếm nhị phân

1	3	5	6	10	11	14	25	26	40	41	78
---	---	---	---	----	----	----	----	----	----	----	----

- Chiến lược chia-đế-trị
- Đầu tiên, khóa được so sánh với phần tử trung tâm của danh sách
- Nếu khóa nhỏ hơn, giới hạn tìm kiếm trong nửa đầu của danh sách
- Nếu không, tìm kiếm trên nửa sau của danh sách

Tìm kiếm nhị phân

- Yêu cầu: Danh sách được sắp xếp
- Tương tự tìm kiếm trên từ điển hoặc trang vàng

VD:

- *Tìm khóa 78*

1	3	5	6	10	11	14	25	26	40	41	78
---	---	---	---	----	----	----	----	----	----	----	----

1	3	5	6	10	11	14	25	26	40	41	78
---	---	---	---	----	----	----	----	----	----	----	----

11 <= 78

14	25	26	40	41	78
----	----	----	----	----	----

26 <= 78

40	41	78
----	----	----

41 <= 78

78

78 = 78

Cần 4 thao tác.

Cần bao nhiêu thao tác nếu tìm kiếm tuần tự?

Algorithm

```
int binSearch(int List[], int Target, int Size) {  
    int Mid, Lo = 0, Hi = Size - 1;  
    while (Lo <= Hi) {  
        Mid = (Lo + Hi) / 2;  
        if (List[Mid] == Target)  
            return Mid;  
        else if (Target < List[Mid])  
            Hi = Mid - 1;  
        else  
            Lo = Mid + 1;  
    }  
    return -1;  
}
```

Example

```
#include <stdio.h>
#define NotFound (-1)
typedef int ElementType;

int main( )
{
    static int A[ ] = { 1, 3, 5, 7, 9, 13, 15 };
    int SizeofA = sizeof( A ) / sizeof( A[ 0 ] );
    int i;
    for( i = 0; i < 20; i++ )
        printf( "BinarySearch of %d returns %d\n",
                i, BinarySearch( A, i, SizeofA ) );
    return 0;
}
```

Exercise 7.1

- Cài đặt phiên bản đệ quy của tìm kiếm tuần tự

Big O Notation

- Định nghĩa: Giả sử $f(n)$ và $g(n)$ là các hàm không âm, $f(n)$ có $O(g(n))$ nếu tồn tại $C > 0$ và $N > 0$ sao cho với mọi $n > N$, $f(n) \leq Cg(n)$.
- $f(n)$ tăng không nhanh hơn $g(n)$; vì vậy $g(n)$ là chặn trên của $f(n)$.
- Big-O thể hiện chặn trên của hàm, với n đủ lớn

Độ phức tạp tính toán

- Thể hiện số phép so sánh
- So với kích thước bài toán (kích thước dữ liệu đầu vào)
- Tìm kiếm tuần tự: $O(n)$
- Tìm kiếm nhị phân: $O(\log_2 n)$

Exercise 7.2

- Tạo một mảng có 100 phần tử với giá trị lần lượt từ 1 tới 100
- Nhập vào một số từ bàn phím, thực hiện tìm kiếm nhị phân, "not found" nếu không tìm thấy
- In ra chỉ số của phần tử trung tâm hiện tại tại mỗi bước tìm kiếm. Cuối cùng, in ra số phép so sánh đã thực hiện trong trường hợp tìm thấy.

Gợi ý

- Với mỗi thao tác so sánh:
 - Tăng biến đếm toàn cục

Exercise 7.3

- So sánh thời gian thực hiện của phiên bản đệ quy và không đệ quy của thuật toán tìm kiếm nhị phân

Thứ tự từ điển

- Việc so sánh hai xâu dựa trên thứ tự từ điển
- Thứ tự từ điển:
 - 'a' < 'd', 'B' < 'M'
 - "acerbook" < "addition"
 - "Chu Trong Hien" > "Bui Minh Hai"
- Sử dụng hàm *strcmp()*

Exercise 7.4

- Xây dựng danh bạ điện thoại
- Khai báo cấu trúc có chứa name, telephone number và e-mail address.
- Khai báo mảng chứa tối đa 100 bản ghi
- Nhập 10 địa chỉ từ tệp vào mảng (đã sắp xếp theo thứ tự từ điển của name)
- Yêu cầu người dùng nhập name, in ra chỉ số đầu tiên có name khớp với name nhập vào; in ra "not found" nếu không tìm thấy