

CÁC GIẢI THUẬT TÌM KIẾM

1. Tìm kiếm tuyến tính
2. Tìm kiếm nhị phân

BÀI TOÁN TÌM KIẾM

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	4	1	-1	4	12	9

$$x = 8$$

- Cho danh sách có n phần tử $a_0, a_1, a_2, \dots, a_{n-1}$.
- Để đơn giản trong việc trình bày giải thuật ta dùng mảng 1 chiều a để lưu danh sách các phần tử nói trên trong bộ nhớ chính.
- Tìm phần tử có khoá bằng X trong mảng
 - Giải thuật tìm kiếm tuyến tính (tìm tuần tự)
 - Giải thuật tìm kiếm nhị phân

TÌM KIẾM TUYẾN TÍNH

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	4	1	-1	4	12	9

x = 8

- **Ý tưởng** : So sánh X lần lượt với phần tử thứ 1, thứ 2,...của mảng a cho đến khi gặp được khóa cần tìm, hoặc tìm hết mảng mà không thấy.
- **Các bước tiến hành**
 - Bước 1: Khởi gán $i=0$;
 - Bước 2: So sánh $a[i]$ với giá trị x cần tìm, có 2 khả năng
 - + $a[i] == x$ tìm thấy x. Dừng;
 - + $a[i] != x$ sang bước 3;
 - Bước 3: $i=i+1$ // Xét tiếp phần tử kế tiếp trong mảng
 - Nếu $i==N$: Hết mảng. Dừng;
 - Ngược lại: Lặp lại bước 2;

TÌM KIẾM TUYẾN TÍNH

- Hàm trả về **true** nếu tìm thấy, ngược lại trả về **false**:

```
boolean LinearSearch(int a[], int x)
{
    int n = a.length;
    int i=0;
    while( (i<n) && ( a[i] != x )) {
        i++;
    }

    if(i==n)
        return false; //Tìm không thấy x
    else
        return true; //Tìm thấy
}
```

TÌM KIẾM TUYẾN TÍNH

^[0] ^[1] ^[2] ^[3] ^[4] ^[5] ^[6]
3 4 1 -1 4 12 9

x = 8

Trường hợp	Số phép so sánh
Tốt nhất	1
Xấu nhất	N
Trung bình	$(N+1) / 2$

➤ Độ phức tạp $O(N)$

TÌM KIẾM NHỊ PHÂN

[0]	[1]	[2]	[3]	[4]	[5]	[6]
-1	1	4	9	12	15	18

$n = 7$
 $x = 15$

➤ **Được áp dụng trên mảng đã có thứ tự.**

➤ **Ý tưởng:**

- Giả sử ta xét mảng có thứ tự tăng, khi ấy ta có $a_{i-1} < a_i < a_{i+1}$
- Nếu $X > a_i$ thì X chỉ có thể xuất hiện trong đoạn $[a_{i+1}, a_{n-1}]$
- Nếu $X < a_i$ thì X chỉ có thể xuất hiện trong đoạn $[a_0, a_{i-1}]$
- Ý tưởng của giải thuật là tại mỗi bước ta so sánh X với phần tử đứng giữa trong dãy tìm kiếm hiện hành, dựa vào kết quả so sánh này mà ta quyết định giới hạn dãy tìm kiếm ở nửa dưới hay nửa trên của dãy tìm kiếm hiện hành.

TÌM KIẾM NHỊ PHÂN

[0]	[1]	[2]	[3]	[4]	[5]	[6]
-1	1	4	9	12	15	18

$n = 7$
 $x = 15$

- Giả sử dãy tìm kiếm hiện hành bao gồm các phần tử nằm trong aleft, aright, các bước của giải thuật như sau:
- Bước 1: left=0; right=N-1;
- Bước 2:
 - $mid = (left + right) / 2$; // chỉ số phần tử giữa dãy hiện hành
 - So sánh a[mid] với x. Có 3 khả năng
 - a[mid] = x: tìm thấy. Dừng
 - a[mid] > x : Right= mid-1;
 - a[mid] < x : Left= mid+1;
- Bước 3: Nếu Left <= Right ; // còn phần tử trong dãy hiện hành
 - + Lặp lại bước 2
 - Ngược lại : Dừng

TÌM KIẾM NHỊ PHÂN

[0]	[1]	[2]	[3]	[4]	[5]	[6]
-1	1	4	9	12	15	18

n = 7
x = 15

➤ Hàm trả về giá trị **true** nếu tìm thấy, ngược lại hàm trả về giá trị **false**

```
Boolean BinarySearch(int a[],int n,int x)
{
    int left, right, mid;
    int left = 0, right = n-1;
    do{
        mid = (left+right)/2;
        if( a[mid] == x )
            return true;
        else if( a[mid]<x )
            left = mid+1;
        else right=mid-1;
    } while(left <= right);
    return false;
}
```



TÌM KIẾM NHỊ PHÂN

[0]	[1]	[2]	[3]	[4]	[5]	[6]
-1	1	4	9	12	15	18

$n = 7$
 $x = 15$

Trường hợp	Số phép so sánh
Tốt nhất	1
Xấu nhất	$\log_2 N$
Trung bình	$\log_2(N / 2)$

➤ Độ phức tạp $O(\log N)$

LUYỆN TẬP TÌM KIẾM - BÀI 1

Viết hàm tìm giá trị và chỉ số của phần tử dương đầu tiên trong mảng 1 chiều các số thực. Mảng được tạo ngẫu nhiên trong đoạn $[-100, 100]$

[0]	[1]	[2]	[3]	[4]	[5]	[6]
-1	-9	0	-3	-11	0	-7

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

LUYỆN TẬP TÌM KIẾM - BÀI 2

Tìm và liệt kê số âm nhỏ nhất trong mảng

[0]	[1]	[2]	[3]	[4]	[5]	[6]
1	8	-2	-9	12	-9	-6

- * Mảng toàn số dương ?
- * Thuật toán tìm min số âm trong mảng:
 1. Tìm số âm đầu tiên -> Gán chỉ số tại đó là **indexMin** là số âm nhỏ nhất
 2. Duyệt phần tử từ **indexMin** + 1 đến cuối mảng
 3. Nếu tìm thấy phần tử âm nhỏ hơn phần tử tại **indexMin** -> cập nhật lại **indexMin**
 4. Duyệt lại mảng và in ra các số âm nhỏ nhất bằng nhau

LUYỆN TẬP TÌM KIẾM - BÀI 3

Tìm số chẵn cuối cùng trong mảng 1 chiều các số nguyên. Mảng được tạo ngẫu nhiên trong đoạn $[-100, 100]$



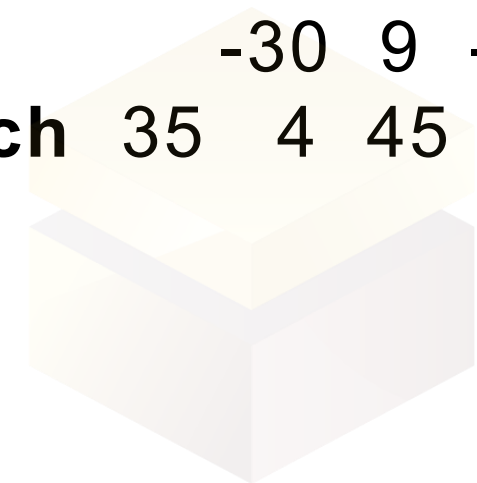
LUYỆN TẬP TÌM KIẾM - BÀI 4

Hãy tìm giá trị & chỉ số tại đó trong mảng các số thực xa giá trị x nhất. x nhập vào từ bàn phím. Mảng được tạo ngẫu nhiên trong đoạn $[-100,100]$

-30 9 -40 12 55 6 9 -60 - Ví dụ $x = 5$?

$x = 5 \Rightarrow$ Khoảng cách

-30	9	-40	12	55	6	9	-60
35	4	45	7	50	1	4	65



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

5

-60 -40 -30 -40 6 9 12 55

LUYỆN TẬP TÌM KIẾM - BÀI TẬP 5

Cho mảng 1 chiều các số thực, hãy tìm đoạn $[a, b]$ sao cho đoạn này chứa tất cả các giá trị trong mảng



LUYỆN TẬP TÌM KIẾM - BÀI 6

Đếm số lần xuất hiện của giá trị x trong mảng



BÀI TẬP NỘP -TÌM KIẾM

1. Tìm 1 vị trí mà giá trị tại vị trí đó là giá trị nhỏ nhất trong mảng 1 chiều các số thực
2. Tìm vị trí của giá trị chẵn đầu tiên trong mảng 1 chiều các số nguyên. Nếu mảng không có giá trị chẵn thì sẽ trả về -1
3. Hãy tìm vị trí giá trị dương nhỏ nhất trong mảng 1 chiều các số thực. Nếu mảng không có giá trị dương thì trả về -1
4. Tìm số nguyên tố cuối cùng trong mảng 1 chiều các số nguyên. Nếu mảng không có số nguyên tố thì trả về -1
5. Hãy tìm giá trị trong mảng các số thực gần giá trị x nhất. x nhập từ bàn phím
6. Cho mảng 1 chiều các số thực, hãy tìm giá trị x sao cho đoạn $[-x, x]$ chứa tất cả các giá trị trong mảng
7. Cho mảng 1 chiều các số nguyên, hãy tìm giá trị đầu tiên nằm trong khoảng $[x, y]$ cho trước. Nếu mảng không có giá trị thỏa điều kiện trên thì trả về -1
8. Cho mảng 1 chiều các số nguyên. Hãy viết hàm tìm giá trị đầu tiên trong mảng có dạng 2^k . Nếu mảng không có giá trị dạng 2^k thì hàm sẽ trả về 0
9. Cho mảng 1 chiều các số nguyên. Hãy viết hàm tìm số nguyên tố nhỏ nhất lớn hơn mọi giá trị có trong mảng
10. Cho mảng 1 chiều các số thực, hãy tìm chỉ số & giá trị đầu tiên lớn hơn giá trị 2020. Nếu mảng không có giá trị thỏa điều kiện trên thì trả về -1