

# THUẬT TOÁN TÌM KIẾM NHỊ PHÂN BINARY SEARCH

# THUẬT TOÁN TÌM KIẾM NHỊ PHÂN

1. Bài toán mở đầu tìm số – trò chơi đoán số
2. Khái niệm về thuật toán
3. Cài đặt thuật toán
4. Bài tập áp dụng

# BÀI TOÁN MỞ ĐẦU

## Tìm giá trị cho trước trong một dãy đã sắp xếp

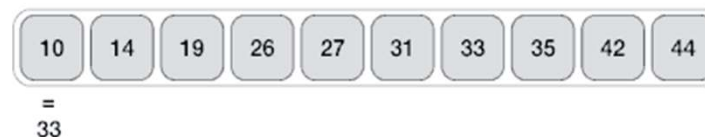
Cho trước một mảng  $A$  có  $n$  ( $0 < n \leq 10^7$ ) phần tử khác nhau, phần tử  $A_i$  có trị tuyệt đối  $\leq 10^9$ . Các phần tử của mảng  $A$  được sắp xếp tăng dần. Yêu cầu: trả về vị trí của phần tử có giá trị  $x$  trong  $A$ .

Ví dụ: Cho  $A = \{0, 5, 13, 19, 22, 41, 55, 68, 72, 81, 98\}$ ,  $x = 55$

Để tìm kiếm vị trí xuất hiện của  $x$  trong mảng  $A$  ta có thể áp dụng:

Thuật toán tìm kiếm tuần tự: Duyệt từ đầu đến cuối mảng  $A$ , nếu tìm thấy phần tử có giá trị bằng với  $x$  thì xuất ra vị trí và ngừng quá trình duyệt. Độ phức tạp tối đa sẽ là  $O(n)$  nếu giá trị phần tử  $x$  cần tìm nằm ở cuối dãy  $A$ .

Linear Search



# TRÒ CHƠI ĐOÁN SỐ

Minh đang nghĩ một con số bí mật trong khoảng từ 1 đến 100. Các em có thể đặt câu hỏi "Số của Minh lớn hơn, nhỏ hơn, hay bằng số X?" để tìm ra số bí mật đó. Làm thế nào để các em tìm ra số bí mật với ít lần đoán nhất?

**Cách chơi tệ:** Đoán lần lượt 1, 2, 3,... Cách này có thể mất tới 100 lần đoán!

**Cách chơi thông minh:**

- Đoán số ở giữa: "Số của Minh có phải là 50 không?"
- Nếu Minh nói "Lớn hơn", em biết ngay số bí mật nằm trong khoảng (50, 100]. Em đã loại bỏ được một nửa các khả năng!
- Bây giờ, em lại đoán số ở giữa khoảng mới:  $(50 + 100) / 2 = 75$ . "Số của Minh có phải là 75 không?"
- Cứ tiếp tục như vậy, mỗi lần đoán em đều loại bỏ một nửa phạm vi tìm kiếm.

Cách chơi thông minh này chính là ý tưởng cốt lõi của **Tìm kiếm nhị phân**. Giống như khi em tìm một từ trong từ điển, em sẽ không lật từng trang từ đầu mà sẽ mở ra khoảng giữa, sau đó quyết định lật về trước hay về sau.

# KHÁI NIỆM

**Tìm kiếm nhị phân (Binary Search)** là một thuật toán tìm kiếm **nhANH**, hoạt động dựa trên nguyên tắc "chia để trị" (divide and conquer).

Điều kiện tiên quyết bắt buộc: Thuật toán này chỉ hoạt động trên một danh sách (mảng) đã được sắp xếp theo thứ tự (ví dụ: từ nhỏ đến lớn).



# Ý TƯỞNG CHÍNH

**Bước 1:** So sánh giá trị cần tìm với phần tử ở chính giữa danh sách.

**Bước 2:**

- Nếu bằng nhau → tìm thấy, kết thúc quá trình tìm kiếm.
- Nếu nhỏ hơn → ta biết chắc chắn giá trị cần tìm (nếu có) phải nằm ở nửa bên trái của danh sách. Ta bỏ qua nửa bên phải.
- Nếu lớn hơn → Ta biết chắc chắn giá trị cần tìm (nếu có) phải nằm ở nửa bên phải của danh sách. Ta bỏ qua nửa bên trái.

**Bước 3:** Lặp lại bước 1 với nửa danh sách còn lại cho đến khi tìm thấy hoặc không còn gì để chia.

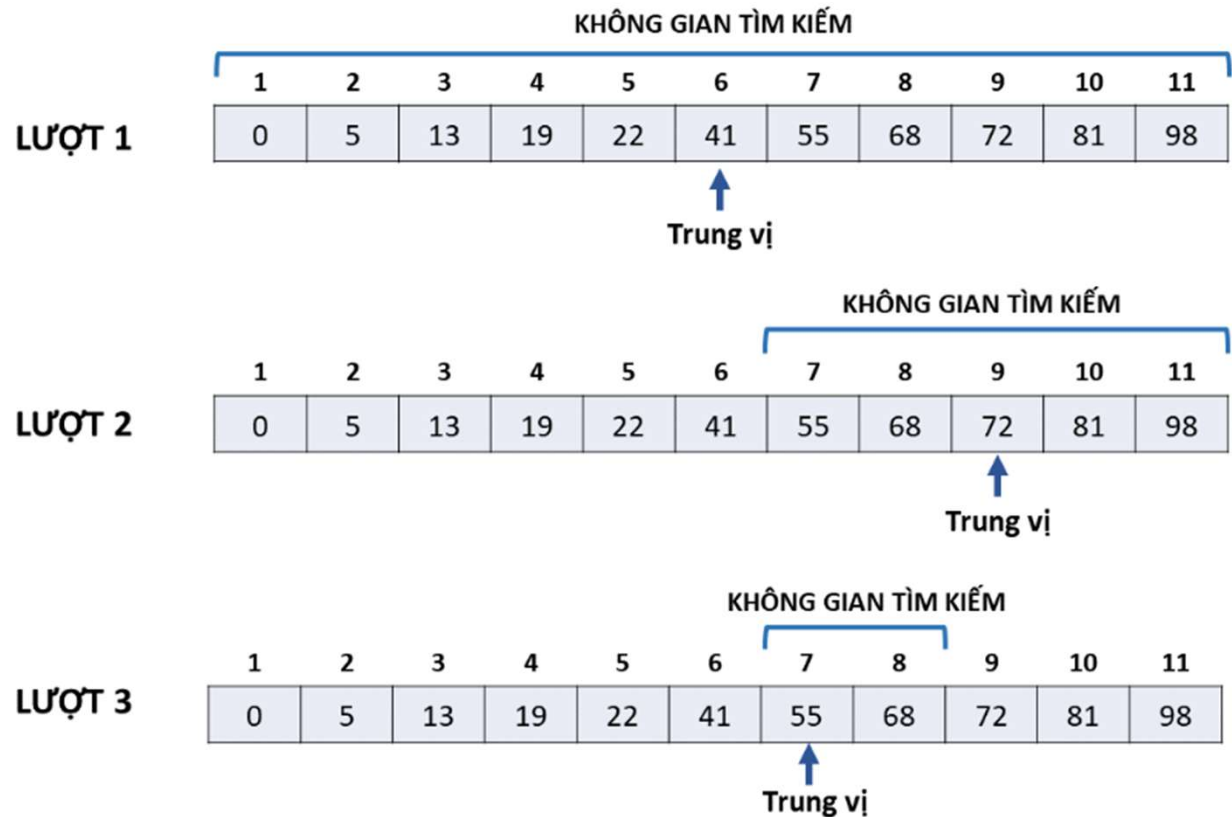
# ÁP DỤNG THUẬT TOÁN

Tìm kiếm nhị phân:

Bước 1: Sắp xếp mảng cần tìm kiếm có thứ tự

Bước 2: Thực hiện quá trình tìm kiếm theo mô tả như hình.

Độ phức tạp thực toán  $O(\log_n)$



# THUẬT TOÁN TÌM KIẾM NHỊ PHÂN

Binary search

steps: 0



1	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Low								mid								high

Sequential search

steps: 0



1	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

[www.mathwarehouse.com](http://www.mathwarehouse.com)

Giáo viên: Hoàng Đăng Quang - 0903781593



# CÀI ĐẶT THUẬT TOÁN

## Hàm tìm kiếm nhị phân mẫu

```
4  int BinarySearch(int A[], int na, int target)
5  {
6      int low = 1, high = na;
7      while (low <= high)
8      {
9          int mid = (lo + high)/2; // mid = low + (high - low)/2;
10         if (A[mid] == target) return mid;
11         else if (A[mid] < target) low = mid + 1;
12         else high = mid - 1;
13     }
14     // Không tìm thấy giá trị trong mảng A thì trả về -1
15     return -1;
16 }
```

# BÀI TẬP ÁP DỤNG

Cho mảng A có n phần tử ( $0 < n \leq 10^7$ ) chứa các phần tử số nguyên  $A_i$  ( $\text{INT\_MIN} \leq A_i \leq \text{INT\_MAX}$ ) theo thứ tự tăng dần và số nguyên X

Tìm vị trí đầu tiên của phần tử X trong mảng A có thứ tự tăng dần.

Tìm vị trí cuối cùng của phần tử X trong mảng A có thứ tự tăng dần.

Tìm vị trí đầu tiên của phần tử lớn hơn X trong mảng A đã sắp tăng dần

Ví dụ:

$A = \{1, 1, 2, 2, 3, 3, 3, 5, 7, 9\}; x = 3$

Vị trí đầu tiên của 3 trong mảng là 5

Vị trí cuối cùng của 3 trong mảng là 7

Vị trí của phần tử đầu tiên lớn hơn 4 trong mảng là 8 và có giá trị là 5