

TÌM KIẾM

Nguyễn Thị Hải Bình
nth.binh@hutech.edu.vn

Mục tiêu

- Hiểu được các giải thuật tìm kiếm và đưa ra hướng giải quyết cho các bài toán tìm kiếm.
- Trình bày và cài đặt các giải thuật tìm kiếm như: tìm kiếm tuyến tính, tìm kiếm nhị phân bằng phương pháp dùng vòng lặp hoặc đệ quy.
- Sử dụng thuật toán tìm kiếm thích hợp để giải quyết các bài toán và các vấn đề trong thực tế.

Nội dung

- Tổng quan về thuật toán tìm kiếm.
- Hai thuật toán tìm kiếm cơ bản:
 - Tìm kiếm tuyến tính (Linear search)
 - Tìm kiếm nhị phân (Binary search)

Tổng quan về thuật toán tìm kiếm

- **Tìm kiếm** = quá trình **xác định** một **đối tượng** nào đó trong một **tập các đối tượng**.
- Kết quả trả về của thuật toán tìm kiếm:
 - Đối tượng tìm được (nếu có).
 - Chỉ số (nếu có) xác định vị trí của đối tượng trong tập đó.
- Việc tìm kiếm dựa theo một trường nào đó của đối tượng.
 - Trường này là khóa (key) của việc tìm kiếm.
 - VD: Tìm sinh viên có họ tên X trong DSSV.
 - SV {MaSV, HoTen, DiaChi,...}
 - Khoá?
 - Kết quả trả về?

Tổng quan về thuật toán tìm kiếm

Các thuật toán tìm kiếm được chia vào **hai nhóm**:

- **Tìm kiếm tuần tự (Sequential Search):**

- Duyệt lần lượt các phần tử trong danh sách, so sánh khóa của phần tử đó với khóa cần tìm.
- Ví dụ: **Tìm kiếm tuyến tính (Linear Search)**.

- **Interval Search:**

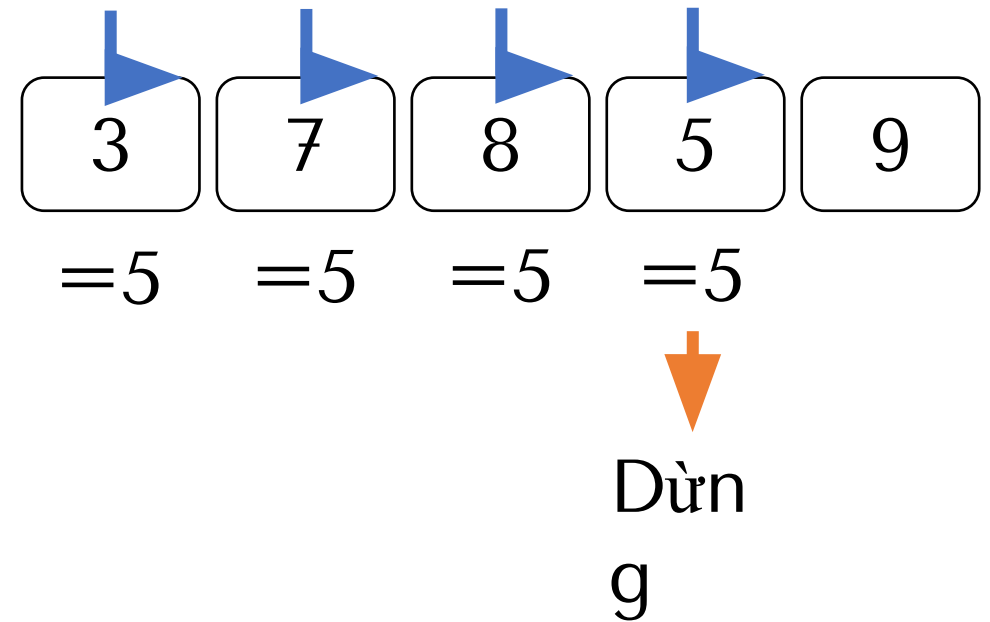
- Các thuật toán thuộc nhóm này thực hiện tìm kiếm trên một tập các đối tượng đã được sắp xếp.
- Hiệu quả hơn so với tìm kiếm tuyến tính.
- Ví dụ: **Tìm kiếm nhị phân (Binary Search)**.

Tìm kiếm tuyến tính (Linear Search)

- Một trong những thuật toán tìm kiếm đơn giản nhất.
- **Duyệt** lần lượt các phần tử trong danh sách, **so sánh** **khóa** của **mỗi phần tử** với khóa cần tìm.
- Nếu **tìm thấy**, thì trả về **vị trí của đối tượng tìm được**. Ngược lại, thuật toán trả về **NULL**.

Ví dụ:

- Tìm vị trí của phần tử có giá trị = 5 trong dãy số: 3, 7, 8, 5, 9.

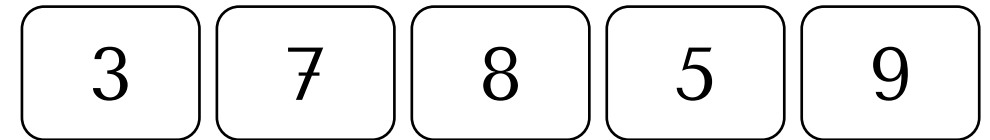


Tìm kiếm tuyến tính (Linear Search)

- Một trong những thuật toán tìm kiếm đơn giản nhất.
- **Duyệt** lần lượt các phần tử trong danh sách, **so sánh** **khóa** của **mỗi phần tử** với khóa cần tìm.
- Nếu **tìm thấy**, thì trả về **vị trí của đối tượng tìm được**. Ngược lại, thuật toán trả về **NULL**.

Example:

- Find the location of 6 in the array 3, 7, 8, 5, 9.



Linear Search Algorithm

1. $i = 0$
2.
 - a) If $a[i] = \text{key}$ return i
 - b) Otherwise, $i = i + 1$
3.
 - a) If $i < n$ Go to step 2
 - b) Otherwise Stop

Linear Search Algorithm

1. $i = 0$
2.
 - a) If $a[i] = \text{key}$ return i
 - b) Otherwise, $i = i + 1$
3.
 - a) If $i < n$ Go to step 2
 - b) Otherwise Stop

Tìm kiếm nhị phân (Binary Search)

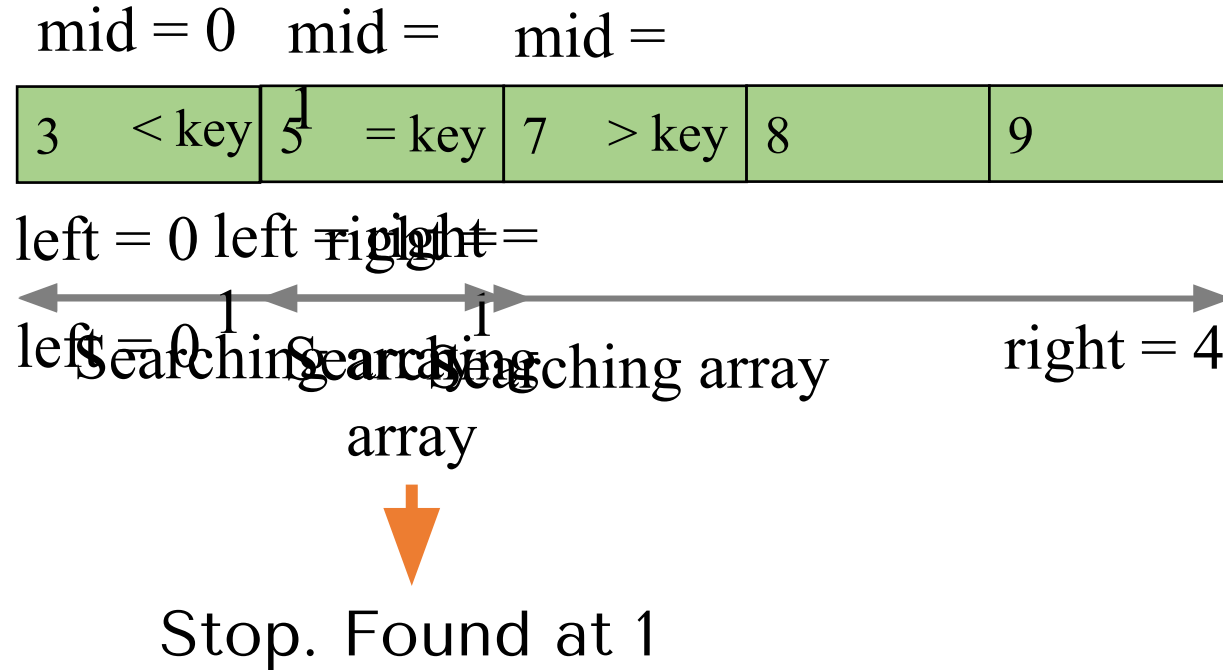
- Hoạt động hiệu quả với **danh sách đã được sắp xếp**.
- Ý tưởng:
 - So sánh khóa của phần tử nằm ở vị trí chính giữa trong danh sách với khóa cần tìm. Nếu trùng nhau thì trả về vị trí chính giữa.
 - Ngược lại, nếu khóa cần tìm có giá trị nhỏ hơn so với phần tử nằm ở vị trí chính giữa, thì tiếp tục tìm kiếm ở nửa bên trái.
 - nếu khóa cần tìm có giá trị lớn hơn so với phần tử nằm ở vị trí chính giữa, thì tiếp tục tìm kiếm ở nửa bên phải.

Binary Search Algorithm

1. Initialize the searching range:
 $\text{left} = 0, \text{right} = n - 1$
2. Divide the searching list into two halves : $\text{mid} = \frac{\text{left} + \text{right}}{2}$
 - a. If $A[\text{mid}] = \text{key} \rightarrow$ return mid.
 - b. If $A[\text{mid}] < \text{key} \rightarrow$ Continue to search on the right part: $\text{left} = \text{mid} + 1$
 - c. If $A[\text{mid}] > \text{key} \rightarrow$ Continue to search on the left part: $\text{right} = \text{mid} - 1$
3. If $\text{left} \leq \text{right} \rightarrow$ Go to step 2
If $\text{left} > \text{right} \rightarrow$ Stop. Not found.

Example:

- Given the below array and key = 5.



Binary Search Algorithm

1. Initialize the searching range:
 $\text{left} = 0, \text{right} = n - 1$
2. Divide the searching list into two halves : $\text{mid} = \frac{\text{left} + \text{right}}{2}$
 - a. If $A[\text{mid}] = \text{key} \rightarrow$ return mid.
 - b. If $A[\text{mid}] < \text{key} \rightarrow$ Continue to search on the right part: $\text{left} = \text{mid} + 1$
 - c. If $A[\text{mid}] > \text{key} \rightarrow$ Continue to search on the left part: $\text{right} = \text{mid} - 1$
3. If $\text{left} \leq \text{right} \rightarrow$ Go to step 2
If $\text{left} > \text{right} \rightarrow$ Stop. Not found.

Example:

- Given the below array and $\text{key} = 6$.

$\text{mid} = 0$ $\text{mid} =$ $\text{mid} =$

3	< key	5	< key	7	> key	8		9
---	-------	---	-------	---	-------	---	--	---

$\text{left} = 0$ $\text{left} =$ $\text{right} =$



$\text{left} = 2$
 $\text{right} = 1$ → Stop. Not found.

Another Example

- Given the below array and key = 8. Show the steps of binary search.

1	2	4	5	6	8	12	15
---	---	---	---	---	---	----	----

Another Example

- Given the below array and key = 2. Show the steps of binary search.

1	2	4	5	6	8	12	15
---	---	---	---	---	---	----	----

Another Example

- Given the below array and key = 0. Show the steps of binary search.

1	2	4	5	6	8	12	15
---	---	---	---	---	---	----	----

Binary Search Algorithm Implementation

1. Initialize the searching range:
 $\text{left} = 0, \text{right} = n - 1$
2. Divide the searching list into two halves : $\text{mid} = \frac{\text{left} + \text{right}}{2}$
 - a. If $A[\text{mid}] = \text{key} \rightarrow$ return mid.
 - b. If $A[\text{mid}] < \text{key} \rightarrow$ Continue to search on the right part: $\text{left} = \text{mid} + 1$
 - c. If $A[\text{mid}] > \text{key} \rightarrow$ Continue to search on the left part: $\text{right} = \text{mid} - 1$
3. If $\text{left} \leq \text{right} \rightarrow$ Go to step 2
If $\text{left} > \text{right} \rightarrow$ Stop. Not found.

Binary Search Algorithm Implementation

1. Initialize the searching range:
 $\text{left} = 0, \text{right} = n - 1$
2. Divide the searching list into two halves : $\text{mid} = \frac{\text{left} + \text{right}}{2}$
 - a. If $A[\text{mid}] = \text{key} \rightarrow$ return mid.
 - b. If $A[\text{mid}] < \text{key} \rightarrow$ Continue to search on the right part: $\text{left} = \text{mid} + 1$
 - c. If $A[\text{mid}] > \text{key} \rightarrow$ Continue to search on the left part: $\text{right} = \text{mid} - 1$
3. If $\text{left} \leq \text{right} \rightarrow$ Go to step 2
If $\text{left} > \text{right} \rightarrow$ Stop. Not found.

Bài tập

Given a list of books. Each book has an identifier (char[10]), title (char[50]), and price (long). Write functions to:

1. Input a list of N books.
2. Find a book by its identifier (using two algorithms: linear and binary).
3. Find a book by its title (using two algorithms: linear and binary).
4. Find all books whose prices are higher than X (X is given by users).
5. Find the most expensive book.