Bài toán tìm kiếm

- Tìm kiếm là quá trình tìm kiếm một phần tử (có thể chỉ là thuộc tính của phần tử) trong tập hợp các phần tử.

Phát biểu bài toán

Input:

- Tập dữ liệu được lưu trữ là dãy phần tử $a_1, a_2,..,a_n$.
- Khoá cần tìm là **key**

Output:

Trả về vị trí tìm thấy key trong dãy (nếu tìm thấy). Nếu không tìm thấy thì trả về -1

CHƯƠNG 2. MỘT SỐ THUẬT TOÁN TÌM KIẾM

Nội dung 01

Nội dung 02

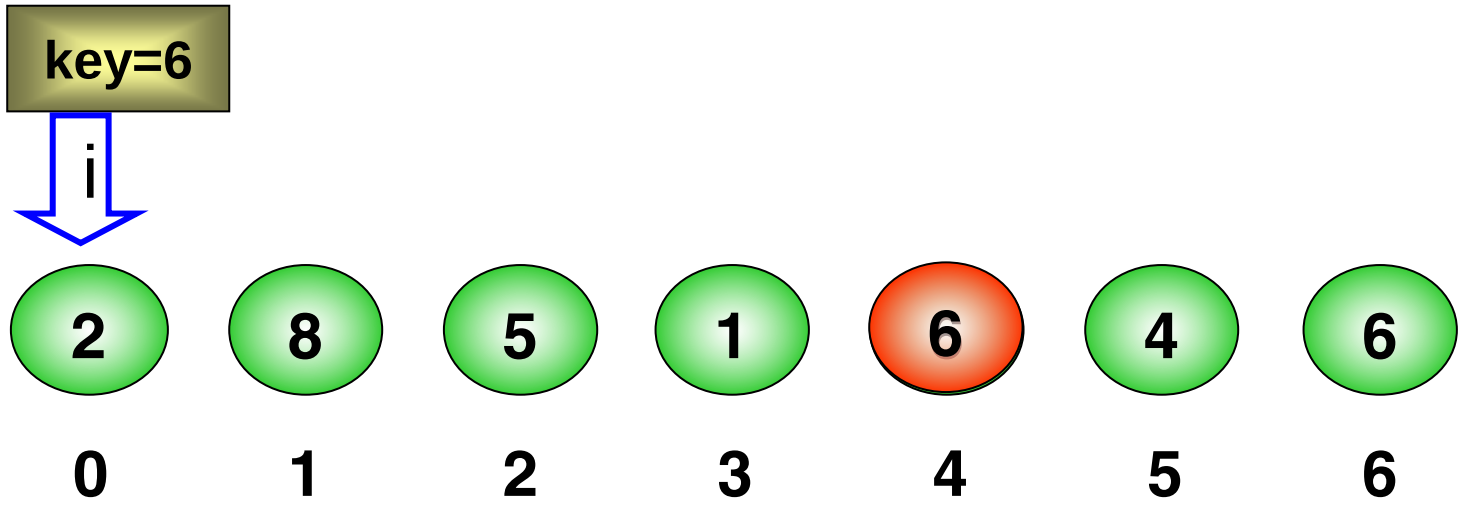
Nội dung 03

Tìm kiếm tuần tự

Mô phỏng

Quan sát mô phỏng từ đó phát biểu lên tư tưởng của giải thuật?

- Khóa cần tìm key = 6

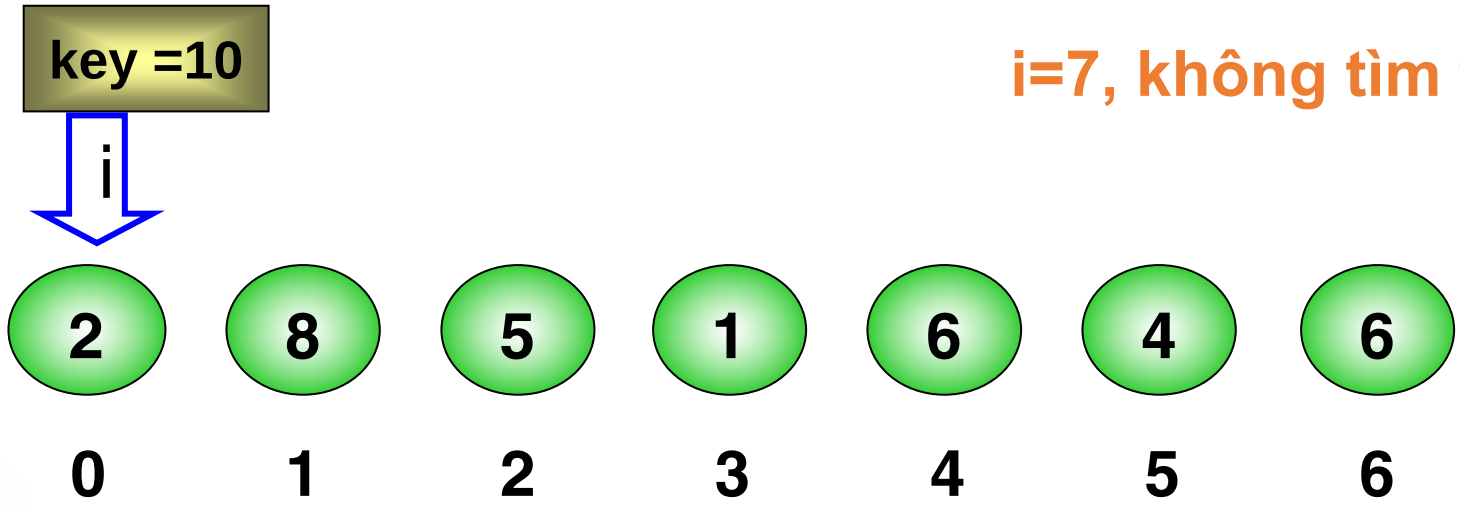


Tìm thấy 6 tại vị trí 4

Mô phỏng

Quan sát mô phỏng từ đó phát biểu lên tư tưởng của giải thuật?

- Khóa cần tìm key = 10



i=7, không tìm thấy

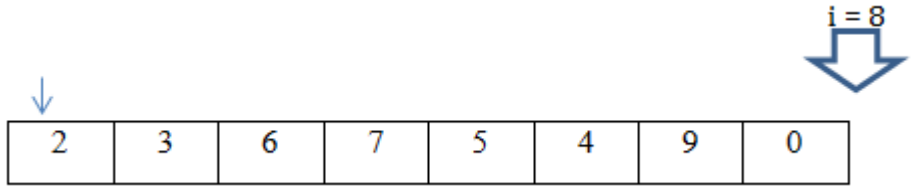
Phát biểu tư tưởng thuật toán

- Bắt đầu từ khoá đầu tiên, lần lượt so sánh khoá key với khoá tương ứng trong dãy.
- Quá trình tìm kiếm kết thúc khi tìm được khoá thoả mãn hoặc đi đến hết dãy hoặc gặp điều kiện dừng vòng lặp

Quá trình phát sinh code

Dựa trên mô phỏng: Yêu cầu 2 sinh viên lên tái hiện thành những đoạn code

Code: với dãy chưa có thứ tự




```
static int UnOrderLinearSearch(int[] arr, int key)
{
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] == key)
            return i;
    }
    return -1;
}
```

Đối với dữ liệu có cấu trúc

```
/*
* Tim kiem sinh vien theo ho ten
*
* Tham so:
        arr: danh sach sinh vien
        n: So luong sinh vien
        key: khoa can tim (ho ten sinh vien can tim)

*/
```

```
static int UnOrderLinearSearch(SinhVien[] arr, string key)
{
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i].hoTen == key)
            return i;
    }
    return -1;
}
```

Code: Khi dãy đã được sắp xếp


```
static int OrderLinearSearch(int[] arr, int key)
{
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] == key)
            return i;
        else if (arr[i] > key)
            break;
    }
    return -1;

}
```

CHƯƠNG 2. MỘT SỐ THUẬT TOÁN TÌM KIẾM

Nội dung 01

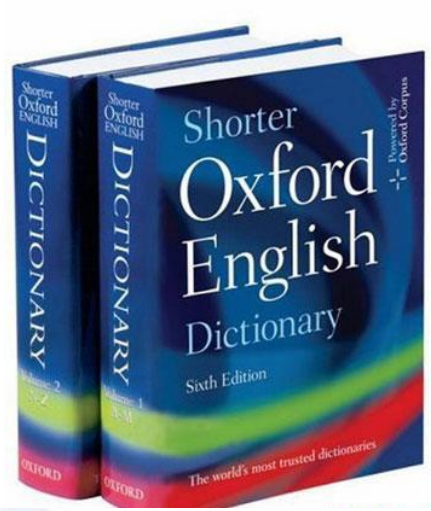
Nội dung 02

Nội dung 03

Tìm kiếm nhị phân

Nêu vấn đề

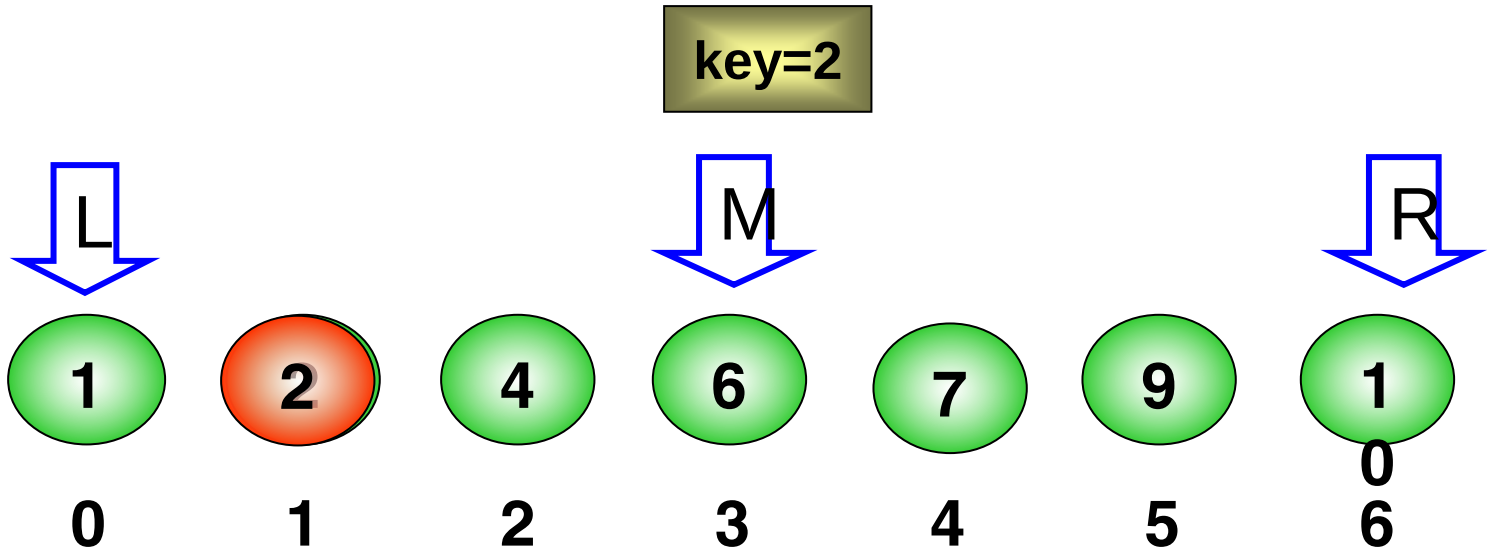
Hãy mô tả quá trình đi tìm một từ mới "Data" trong một cuốn từ điển



Mô phỏng thuật toán

Quan sát mô phỏng từ đó phát biểu lên tư tưởng của giải thuật?

- Khóa cần tìm Key = 2

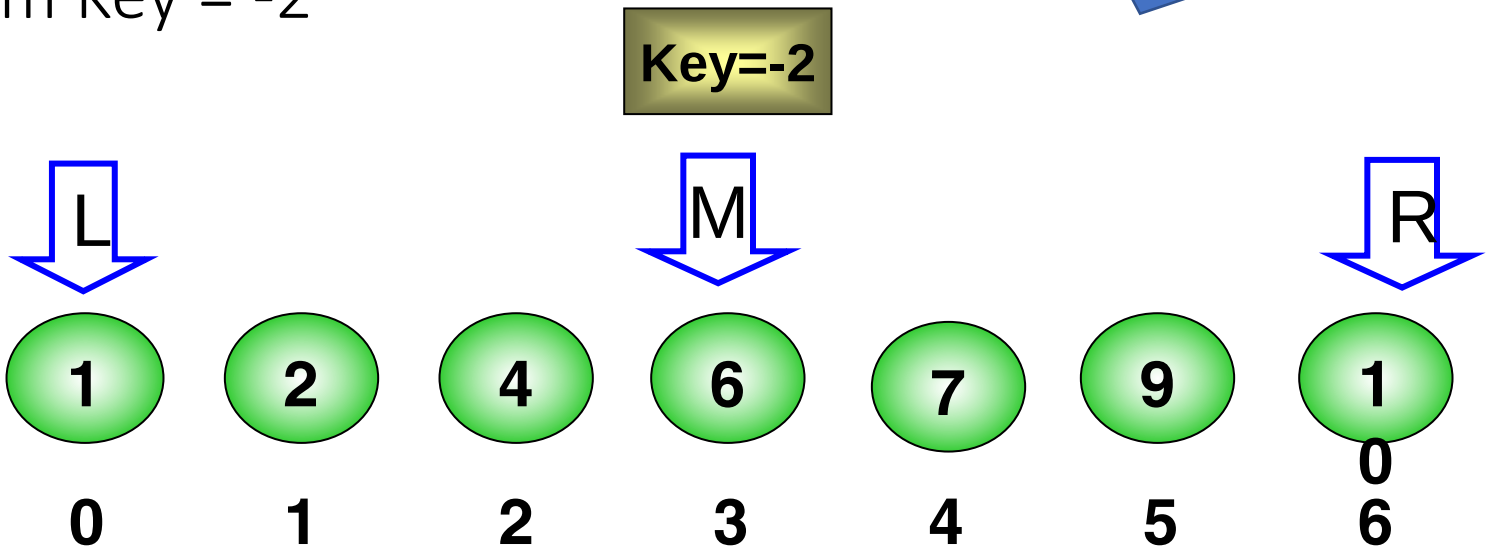


Tìm thấy 2 tại vị trí 1

Mô phỏng thuật toán

Quan sát mô phỏng từ đó phát biểu lên tư tưởng của giải thuật?

- Khóa cần tìm Key = -2



L=0

R=-1 => không tìm thấy Key = -2

Phát biểu tư tưởng thuật toán

- Giải thuật tìm kiếm nhị phân làm việc dựa trên nguyên tắc chia để trị (Divide and Conquer). Chỉ áp dụng cho **dãy đã được sắp xếp**.

Binary Search tìm kiếm một phần tử cụ thể bằng cách so sánh phần tử tại vị trí chính giữa của tập dữ liệu.

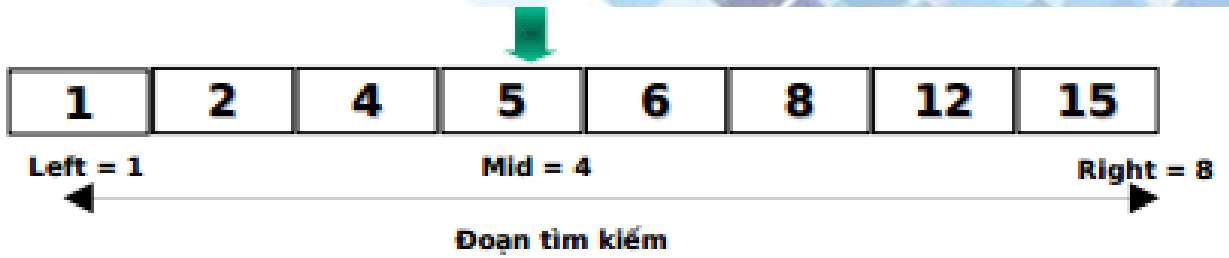
- Nếu tìm thấy thì vị trí của phần tử được trả về.
- Nếu phần tử cần tìm là lớn hơn giá trị phần tử giữa thì phần tử cần tìm được tìm trong mảng con nằm ở bên phải phần tử giữa, ngược lại thì sẽ tìm ở trong mảng con nằm ở bên trái phần tử giữa.
- Tiến trình sẽ tiếp tục như vậy trên mảng con cho tới khi tìm hết mọi phần tử trên mảng con này.

Quá trình phát sinh code

Dựa trên mô phỏng: Yêu cầu 2 sinh viên lên tái hiện thành những đoạn code

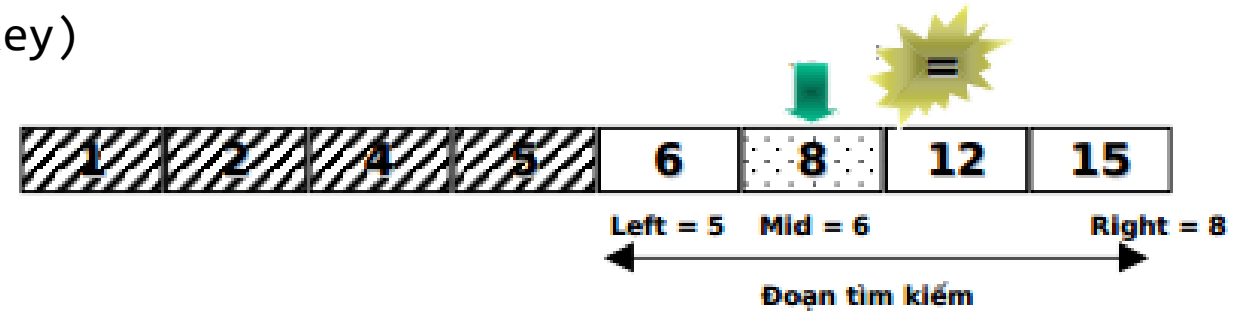
HCMUTE

Code




```
static int BinarySearch (int[] arrInt, int key)
{
    int left = 0;
    int right = arrInt.Length - 1;
    int mid;
    while (left <= right)
    {
            mid = (left + right) / 2;
            if (arrInt[mid] == key)
                    return mid;
            else if (arrInt[mid] < key)
                    left = mid + 1;
            else right = mid - 1;
    }
    return -1;

}
```



HCMUTE

Nhận xét

- Tìm kiếm tuần tự: O(n)
- Tìm kiếm nhị phân: O(logn)
- Khi dữ liệu đã được sắp xếp: Thì thuật toán tìm kiếm nhị phân tối ưu hơn thuật toán tìm kiếm tuần tự, vì mỗi lần tìm kiếm số bước đã được giảm đi một nửa.
- Khi dữ liệu chưa được sắp xếp: Thì thuật toán tìm kiếm tuần tự tốt hơn vì không bỏ chi phí sắp xếp và sau mỗi lần lặp số phần tử cần xét giảm 1/2.

Tìm kiếm trên dữ liệu có cấu trúc

- Có nhận xét gì kiểu dữ liệu của khóa tìm kiếm và kiểu dữ liệu của danh sách?

Củng cố bài học

Trò chơi điền khuyết

- Hãy điền khuyết vào từng vị trí bị che khuất để hoàn thành giải thuật?

```
static int UnOrderLinearSearch(int[] arr, ir
{
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] =
            return
    }
    retur
}
```

1

2

3

Củng cố bài học

Trò chơi điền khuyết

- Hãy điền khuyết vào từng vị trí bị che khuất để hoàn thành giải thuật?

```
static int OrderLinearSearch(int[] arr, int key)
{
    for (int i = 0; i < arr.Length; i++)
    {
        if (arr[i] =
            return
        else if (arr
            break;
    }
    retu

}
```

1

2

3

Củng cố bài học

- Hãy điền khuyết vào từng vị trí bị che khuất để hoàn thành giải thuật?

```
static int BinarySearch (int[] arrI
{
    int
    int
    int mid;
    while (left
    {
                r
                i
                e
                e
    }
    return -1;

}
```

1

2

3

4

Trò chơi điền khuyết