

# TÌM KIẾM NHỊ PHÂN

➤ Điều kiện tiên quyết để áp dụng thuật toán tìm nhị phân là các đối tượng trong “không gian tìm kiếm” phải được sắp xếp thứ tự theo một tiêu chuẩn nào đó và ta sẽ dựa trên tiêu chuẩn này để tìm.

## TƯ TƯỞNG THUẬT TOÁN

➤ Thuật toán tìm nhị phân chia không gian tìm kiếm thành 2 không gian con. Sau đó quyết định tìm kiếm tiếp 1 trong 2 không gian con này dựa vào tiêu chuẩn tìm kiếm.

# VD: Cho mảng được sắp tăng như sau

0	1	2	3	4	5	6	7	8	9
-20	-3	6	12	32	45	86	90	91	100

Tìm  $x = 45$

```
l=0
r=9
l<=r Đ
    mid=(0+9)/2=4
    a[mid]=32<x=45
    l=mid+1=5 (r=9)
l<=r Đ
    mid=(5+9)/2=7
    a[mid]=90>x
    r=mid-1=6 (l=5)
l<=r Đ
    mid=(5+6)/2=5
    a[mid]=45==x
    return mid(5)
```

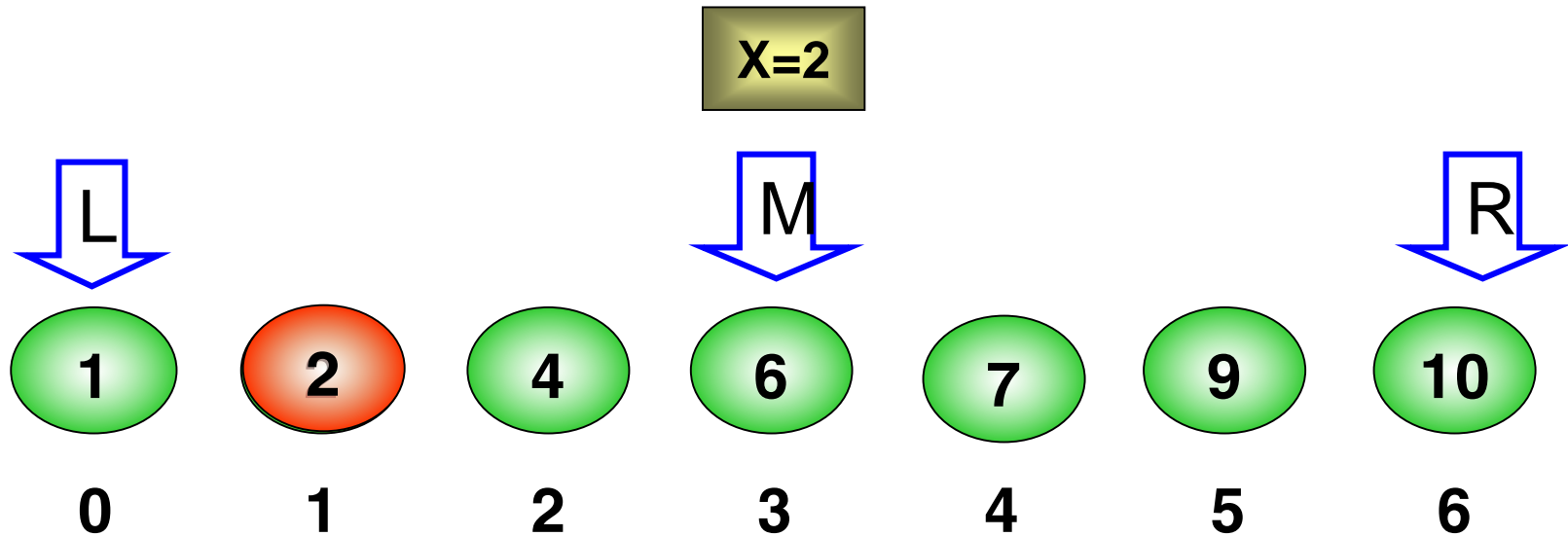
Tìm  $x = 46$

```
l=0 r=9
l<=r Đ
    mid=(0+9)/2=4
    a[mid]=32<x=46
    l=mid+1=5 (r=9)
l<=r Đ
    mid=(5+9)/2=7
    a[mid]=90>x=46
    r=mid-1=6 (l=5)
l<=r Đ
    mid=(5+6)/2=5
    a[mid]=45<x=46
    l=mid+1=6 (r=6)
l<=r Đ
    mid=(6+6)/2=6
    a[mid]=86>x=46
    r=mid-1=5 (l=6)
l>r
return -1
```

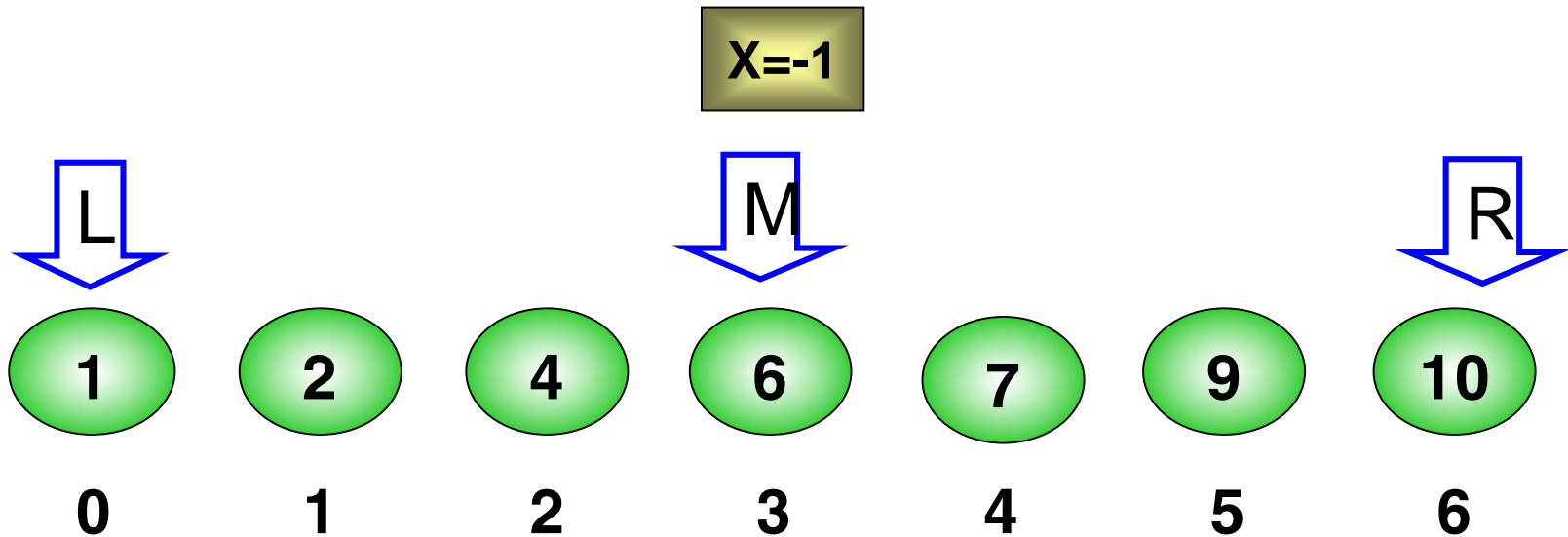
ThS.Nguyễn Thúy Loan

# Minh Họa Thuật Toán Tìm Nhị Phân

Tìm thấy 2 tại vị trí 1



# Minh Họa Thuật Toán Tìm Nhị Phân (tt)



$L=0$

$R=-1 \Rightarrow$  không tìm thấy  $X=-1$

# Các Bước Thuật Toán Tìm Kiếm Nhị Phân

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

# TÌM KIẾM NHỊ PHÂN

Bài toán: tìm vị trí giá trị x trong mảng một chiều các số thực được sắp thứ tự tăng dần.

```
int BinarySearch ( float a[], int n, float x)
{
    int l=0;
    int r=n-1;
    while(l<=r)
    {
        int mid = (l+r)/2;
        if (a[mid]==x)           return mid;
        else
            if (a[mid]>x)         r=mid-1;
            else                  l=mid+1;
    }
    return -1;
}
```

# Đánh giá độ phức tạp của thuật toán TKNP

Trường hợp	Css
Tốt nhất	1
Xấu nhất	$\log_2 N$
Trung bình	$\log_2 N / 2$

➤ Độ phức tạp  $O(\log_2 N)$

# TÌM KIẾM NHỊ PHÂN

Khai báo cấu trúc dữ liệu cho 1 sản phẩm gồm :

1. *MãSP* ( số nguyên 4 byte )
2. *TênSP* ( chuỗi tối đa 20 kí tự )
3. *GiáSP* ( số thực )

Giả sử đã có mảng 1 chiều *a* gồm *n* sản phẩm ( $0 < n < 100$  )

1. Viết hàm nhập xuất
2. Sắp xếp mảng tăng dần theo MãSP .
3. Tìm sản phẩm có mãSP là *x* ( bằng 2 PP )
4. Viết hàm thêm 1 sản phẩm mới vào mảng sao cho vẫn giữ nguyên thứ tự tăng của mãSP .

ThS.Nguyễn Thúy Loan