



**FPT POLYTECHNIC**



Bài 5:  
CÁC THUẬT TOÁN  
SỐ PHẪNG VÀ TÌM KIẾM

---

[www.poly.edu.vn](http://www.poly.edu.vn)

[hoclaptrinhweb.com](http://hoclaptrinhweb.com)

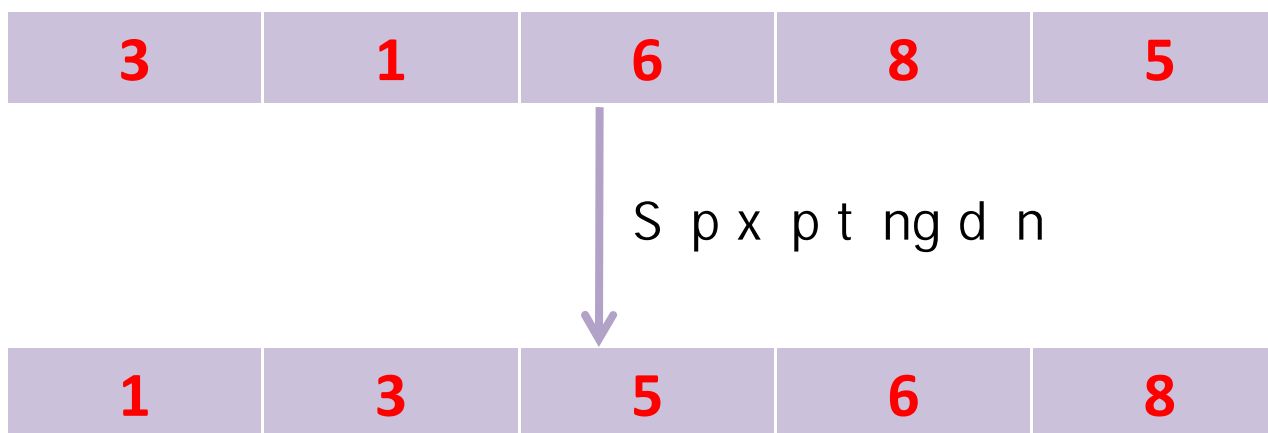
- Tìm hi u v cách s d ng m ng thông th ng trong VB.Net
- Tìm hi u v l p ArrayList và cách s d ng trong VB.Net
- So sánh m ng thông th ng và ArrayList
- Áp d ng vi c o th i gian th c hi n l nh

## Mục tiêu bài học hôm nay

- Tìm hiểu các ghi chú toán học và tìm kiếm trên cấu trúc dữ liệu mảng
- Tìm hiểu các ghi chú toán học và tìm kiếm trên cấu trúc dữ liệu mảng
- Đánh giá và so sánh hiệu quả các ghi chú toán học

## nh ngh a bài toán s p x p

- S p x p là quá trình x lý m t danh sách các ph n t t chúng theo m t th t nào ó (t ng d n, gi m d n) d a trên n i dung thông tin l u gi t i m i ph n t .



## Bài toán sắp xếp dãy số

- **Bài toán:** Cho trước một dãy số  $a_1, a_2, \dots, a_N$  có  
lưu trữ trong cấu trúc dữ liệu mảng
- **Sắp xếp dãy số**  $a_1, a_2, \dots, a_N$  là thể hiện vị trí của  
các phần tử sao cho hình thành nên dãy mới  $a_{k1}, a_{k2},$   
 $\dots, a_{kN}$  có thể (ví dụ thể hiện) nghĩa là  $a_{ki} > a_{ki-1}$ .

## Bài toán sắp xếp dãy số

- quy tắc nhúng các nhúng tình huống cần thay đổi vị trí các phần tử trong dãy, cần dựa vào kết quả của một loạt phép so sánh
- > Hai thao tác **so sánh** và **gán** là các thao tác cơ bản cấu thành thuật toán sắp xếp.
- Chú ý: Khi xây dựng một thuật toán sắp xếp cần tìm cách giảm thiểu nhu cầu phép so sánh và nếu có thể thì tận dụng hiệu quả của thuật toán

## 3 kĩ thuật sắp xếp chọn

- Sắp xếp lựa chọn (Selection Sort)
- Sắp xếp nổi bọt (Bubble Sort)
- Sắp xếp chèn (Insertion Sort)

## ■ Ý tưởng:

- Chọn phần tử nhỏ nhất trong N phần tử ban đầu, đưa phần tử này về vị trí đầu dãy hiên hành; sau đó lặp lại quá trình sắp xếp tiếp theo.
- Xem dãy hiên hành chỉ còn N-1 phần tử của dãy ban đầu, bắt đầu vị trí thứ 2; lặp lại quá trình trên cho dãy hiên hành... đến khi dãy hiên hành chỉ còn 1 phần tử.



# Sắp xếp chèn

- Các bước sắp xếp chèn:

Bước 1:  $i = 1$  // Lấy lý đầu tiên

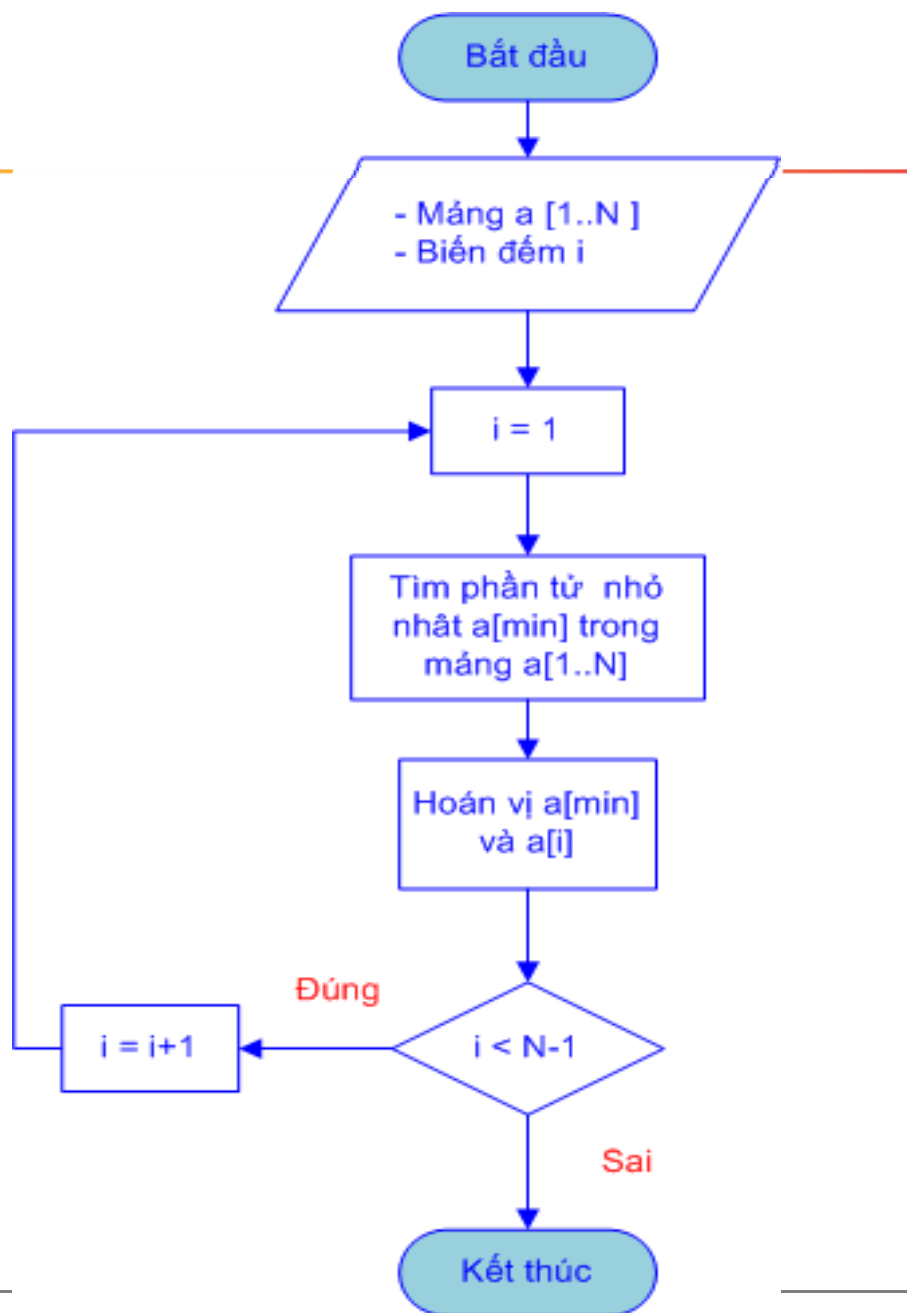
Bước 2: Tìm phần tử nhỏ nhất  $a[\min]$  trong dãy hiện hành từ  $a[i]$  đến  $a[N]$

Bước 3: Hoán vị  $a[\min]$  và  $a[i]$

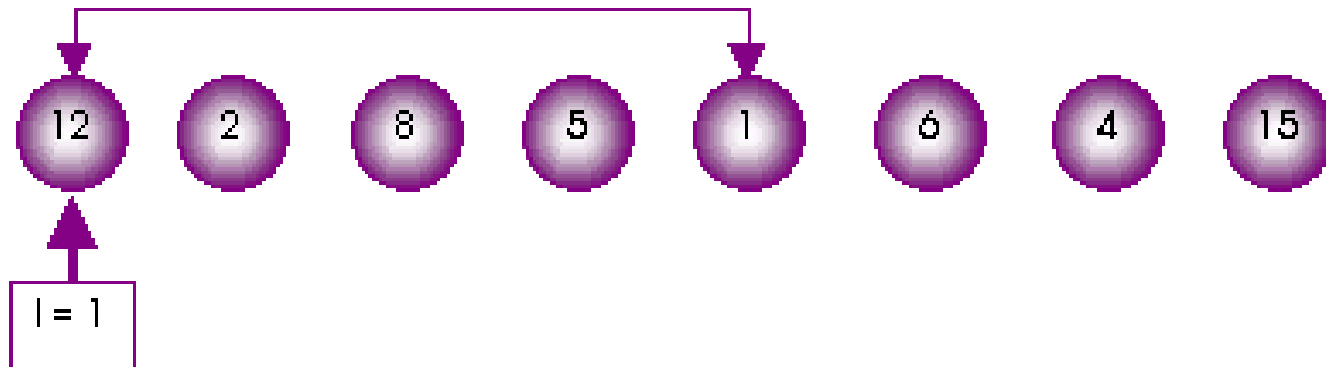
Bước 4: Nếu  $i < N-1$  thì  $i = i+1$ ; Lặp lại Bước 2

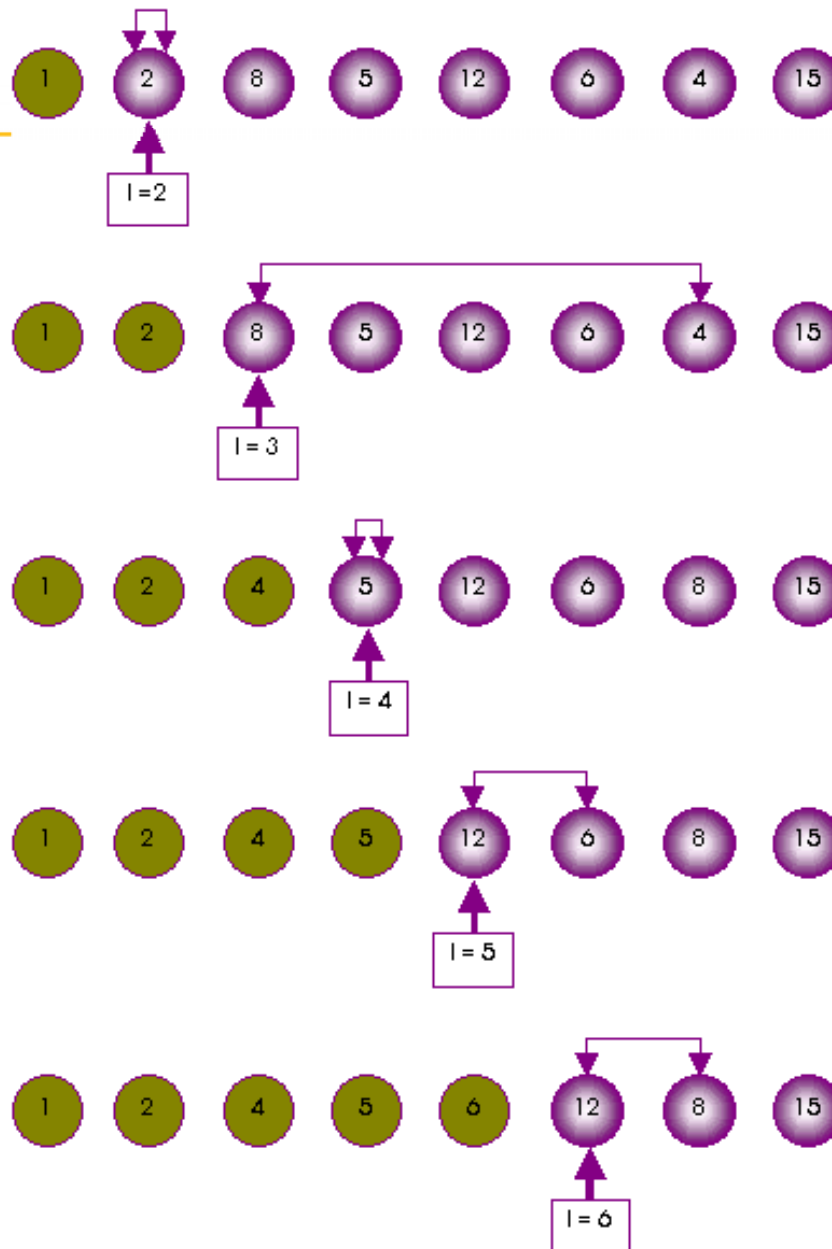
Ngừng

# L U G I I THU T S P X P L A C H N



- Ví dụ : Cho dãy số a:  
 $\{12, 2, 8, 5, 1, 6, 4, 15\}$





- Cài đặt giải thuật sắp xếp chọn bằng ngôn ngữ VB.Net

```
Public Sub SelectionSort()
    Dim outer, inner, min, temp As Integer
    For outer = 0 To numElements - 2
        min = outer
        For inner = outer + 1 To numElements - 1
            If (arr(inner) < arr(min)) Then
                min = inner
            End If
        Next
        ' Hoán vị phần tử nhỏ nhất với phần tử đầu tiên
        temp = arr(outer)
        arr(outer) = arr(min)
        arr(min) = temp
    Next
End Sub
```

# S p x p l a c h \_ n

- ánh giá gi i thu t trên: l t th i, bao gi c ng c n (n-i) l n so sánh xác nh ph n t nh nh t hi n hành. Do v y s l n so sánh:

$$\sum_{i=1}^{n-1} (n-i) = \frac{n(n-1)}{2}$$

- S l n hoán v (m t hoán v b ng 3 phép gán) l i ph thu c vào tình tr ng ban u c a dãy s , ta ch có th c l c trong t ng tr ng h p nh sau:

Tr ng h p	S l n so sánh	S phép gán
T t nh t	$n(n-1)/2$	$3(n-1)$
X u nh t	$n(n-1)/2$	$n(n-1)/2 + 3(n-1)$

# Sắp xếp nổi bọt

- Ý tưởng: xuất phát từ dãy, so sánh 2 phần tử cạnh nhau và phần tử nhỏ hơn lên trên, sau đó lặp lại xét tiếp theo cho đến khi tất cả dãy. Như vậy, lần x lý thứ i tìm được phần tử vị trí cuối dãy là i

- Các bước:

Bước 1:  $i=1$  // lần x lý đầu tiên

Bước 2:  $j=N$  // duy nhất cuối dãy trở về vị trí

Trong khi  $j > i$  thì thực hiện:

Nếu  $a[j] < a[j-1]$ : hoán vị  $a[j]$  và  $a[j-1]$

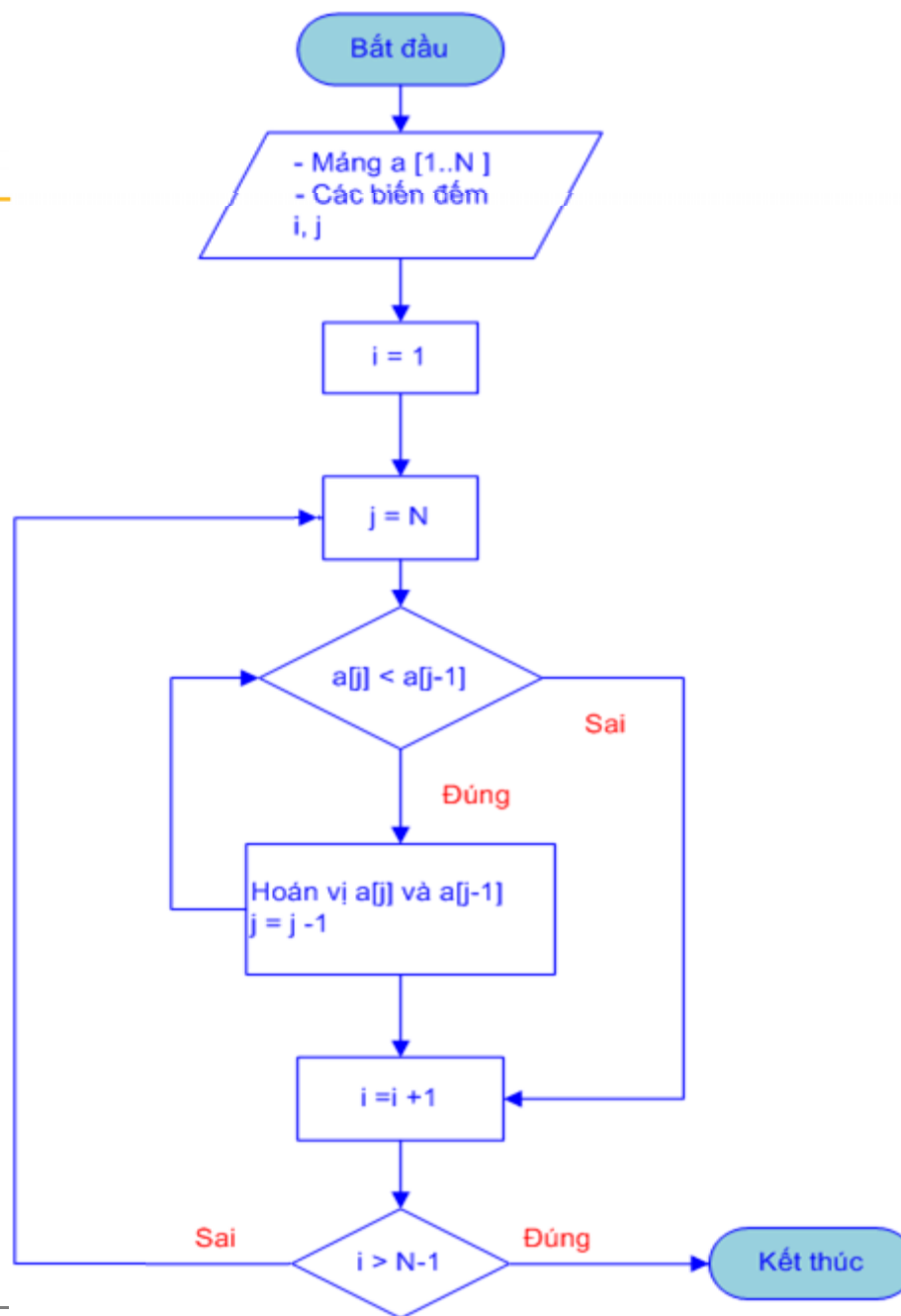
$j = j-1$

Bước 3:  $i=i+1$  // lần x lý tiếp theo

Nếu  $i > N-1$  thì dừng

Ngược lại, lặp lại Bước 2

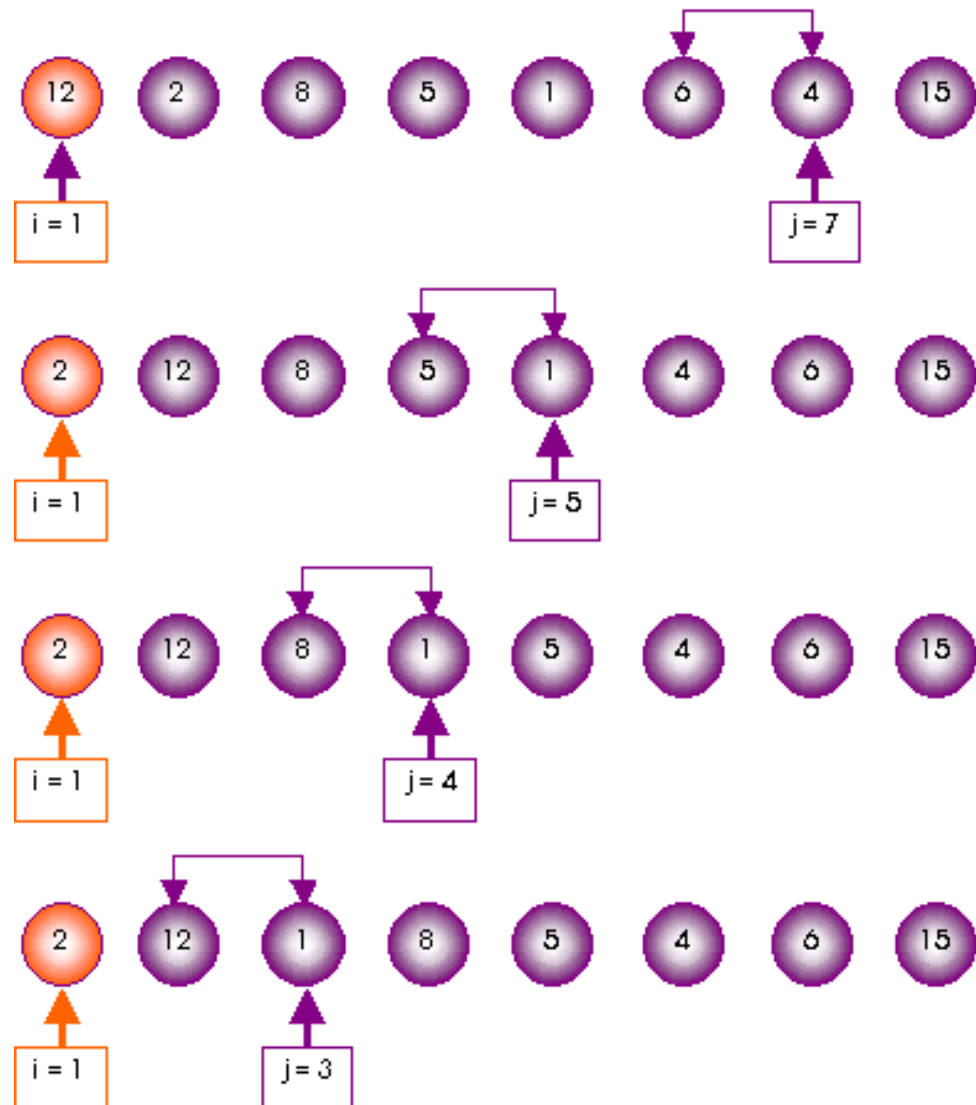
# L U G I I THU T S P X P N I B T

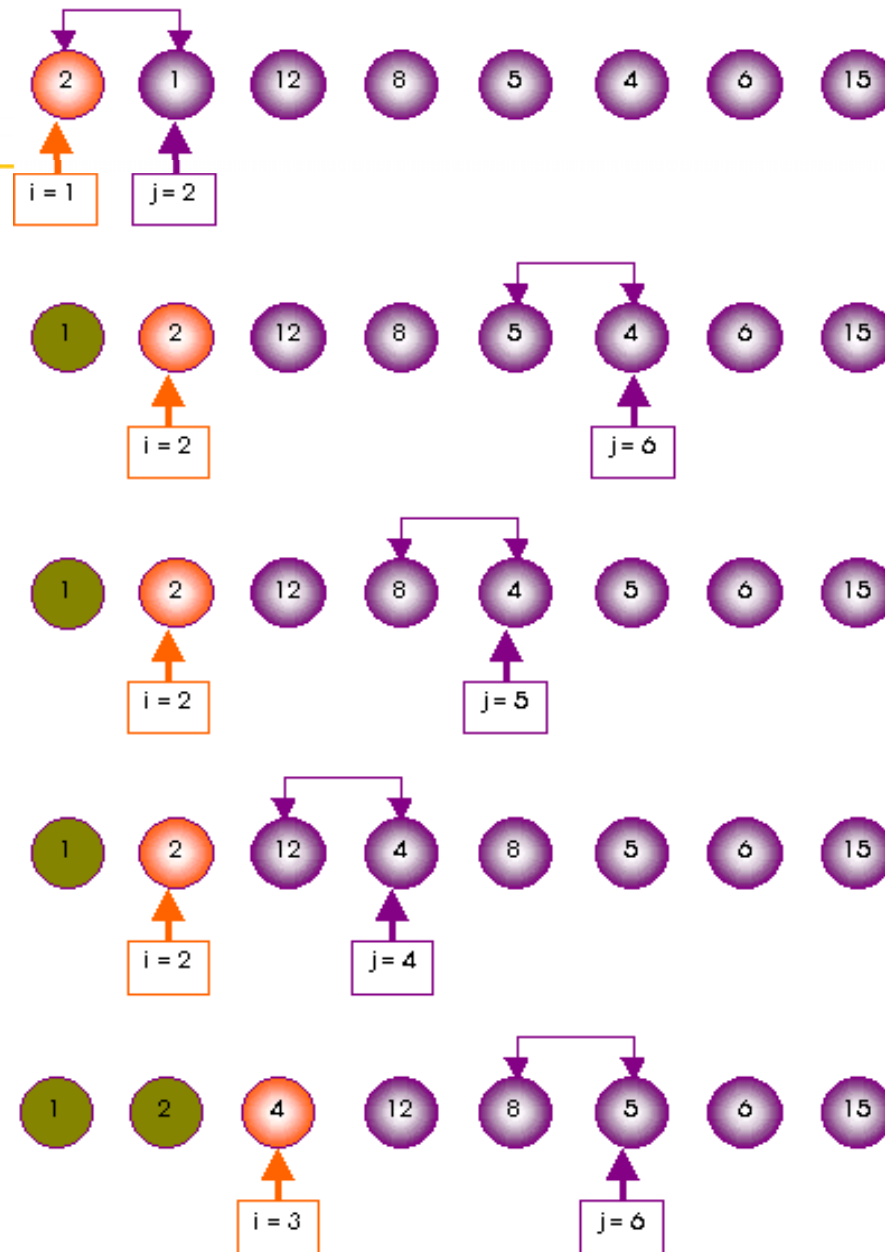


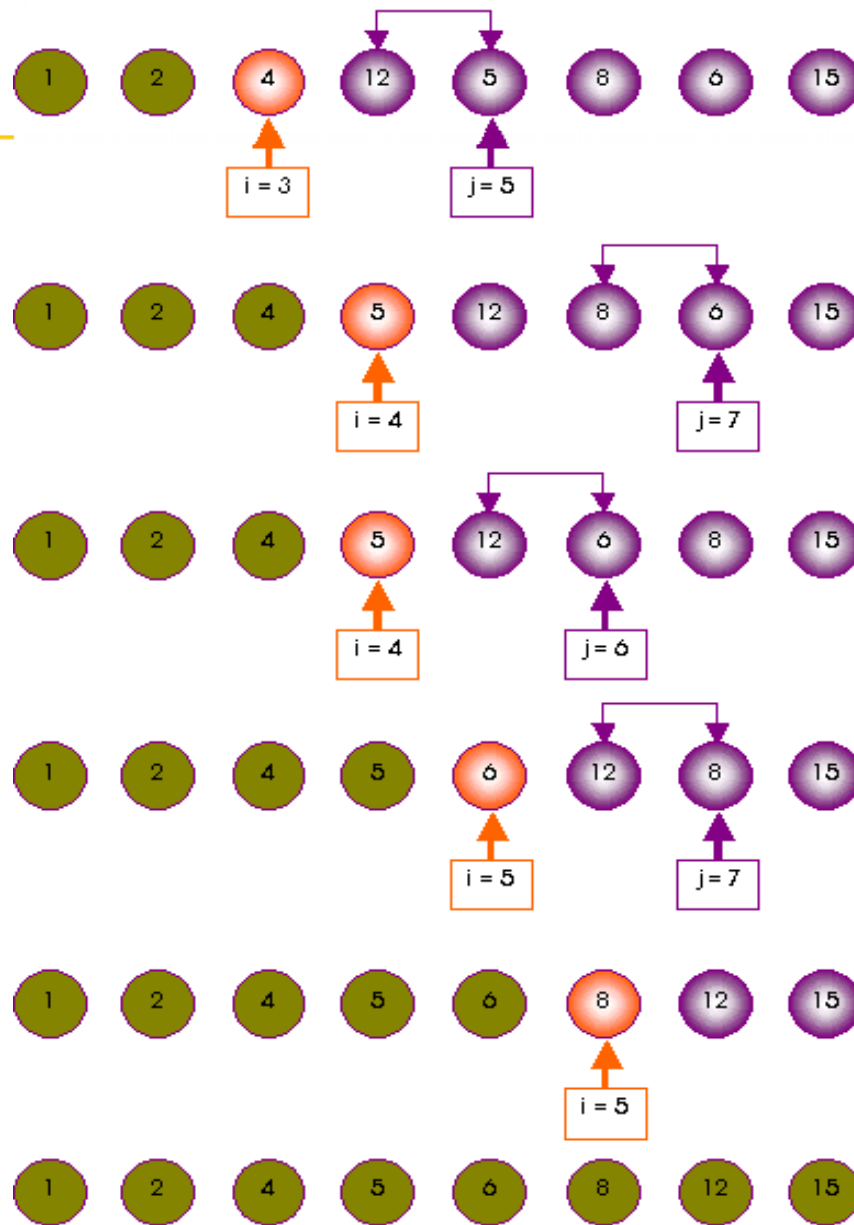


# Sắp xếp nổi bọt

■ Ví dụ : Cho dãy số a:  
 $\{12, 2, 8, 5, 1, 6, 4, 15\}$







- Cài đặt giải thuật sắp xếp ngôn ngữ VB.Net

```
Public Sub BubbleSort()
    Dim outer, inner, temp As Integer
    For outer = numElements - 1 To 2 Step -1
        For inner = 0 To outer - 1
            If (arr(inner) > arr(inner + 1)) Then
                temp = arr(inner)
                arr(inner) = arr(inner + 1)
                arr(inner + 1) = temp
            End If
        Next
    Next
End Sub
```

## ■ ánh giá gi i thu t:

- I t th i, bao gi c ng c n  $(n-i+1)$  I n so sánh xác nh ph n t nh nh t hi n hành. Do v y s I n so sánh:

$$\sum_{i=1}^{n-1} (n-i+1) = \frac{n(n-1)}{2}$$

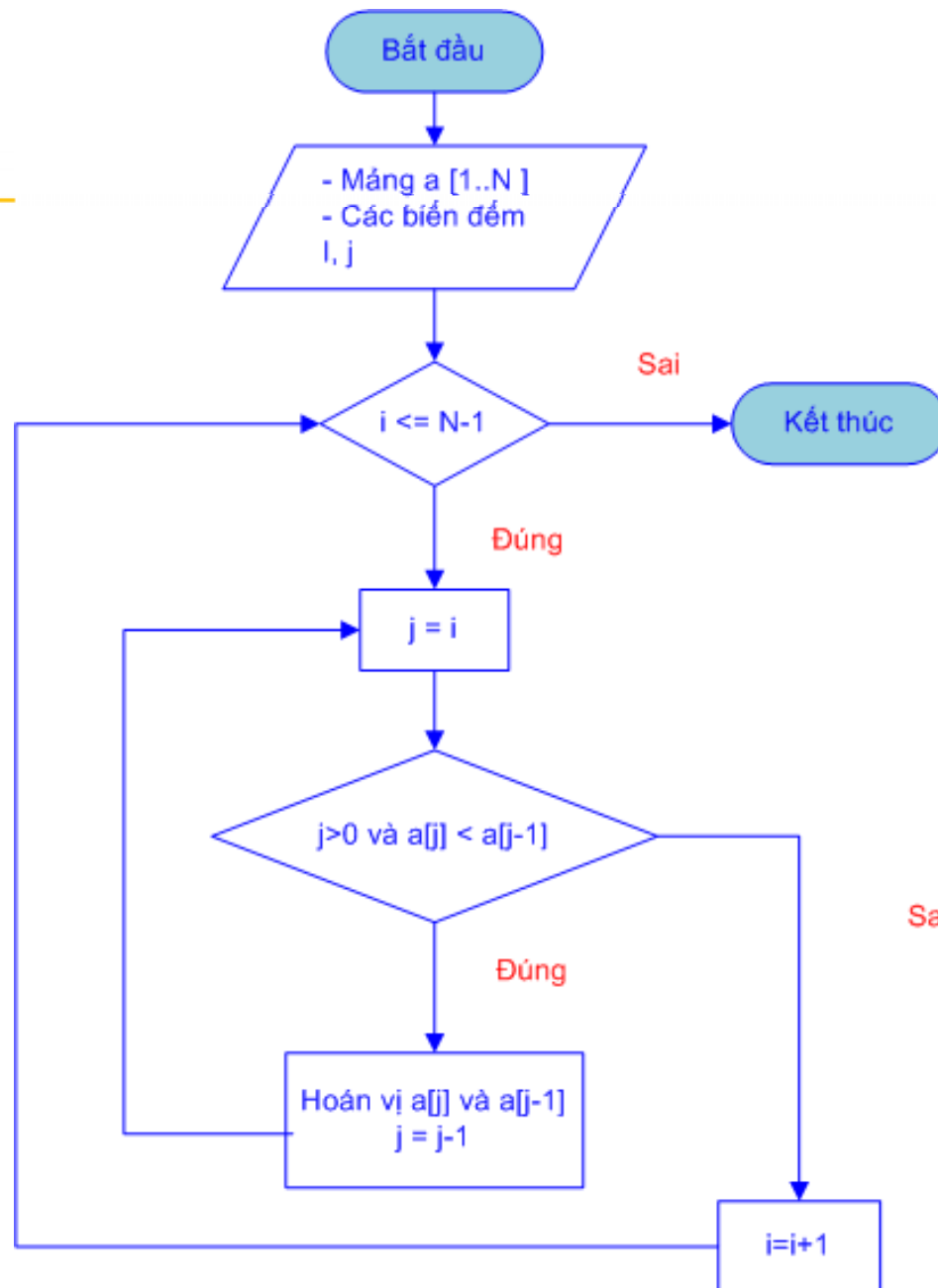
- S I ng phép hoán v th c hi n tùy thu c vào k t qu so sánh, có th c I c trong t ng tr ng h p nh sau

Tr ng h p	S l n so sánh	S I n hoán v
T t nh t	$\frac{n(n-1)}{2}$	0
X u nh t	$\frac{n(n-1)}{2}$	$\frac{n(n-1)}{2}$

## ■ Ý tưởng:

- Trước hết ta xem phần tử  $a[0]$  là một dãy đã có thứ tự.
- Bước 1: Chèn phần tử  $a[1]$  vào đúng vị trí trong dãy  $a[0]$  trên sao cho dãy gồm  $a[0]$  và  $a[1]$  sắp xếp thứ tự.
- Bước 2: Chèn phần tử  $a[2]$  vào đúng vị trí trong dãy gồm  $a[0]$ ,  $a[1]$  sao cho dãy gồm  $a[0]$ ,  $a[1]$ ,  $a[2]$  sắp xếp thứ tự.
- Tổng quát bước  $i$ , chèn phần tử  $a[i]$  vào đúng vị trí trong dãy đã sắp xếp  $a[0], \dots, a[i-1]$  sao cho dãy  $a[0], a[1], \dots, a[i]$  sắp xếp thứ tự.
- Sau  $N-1$  bước thì kết thúc.

L U GI I  
THU T S P  
X P CHÈN



# Sắp xếp chèn

- Ví dụ : cho danh sách gồm 7 phần tử , trong đó 3 phần tử đầu tiên đã được sắp xếp

1	3	7	-	6	4	2	5
---	---	---	---	---	---	---	---

- Thực hiện thao tác sắp xếp phần tử  $a_4 = 6$  vào danh sách còn lại, ta tìm vị trí thích hợp để chèn nó là sau 3 và trước 7.

1	3	6	7	-	4	2	5
---	---	---	---	---	---	---	---

- Làm tiếp theo với  $a_5 = 4$  ta có

1	3	4	6	7	-	2	5
---	---	---	---	---	---	---	---



- Làm tiếp theo với  $a_6 = 2$  ta có

1	2	3	4	6	7	-	5
---	---	---	---	---	---	---	---

- Cuối cùng chèn  $a_7 = 5$

1	2	3	4	5	6	7	-
---	---	---	---	---	---	---	---

- Cài đặt giải thuật sắp xếp ngôn ngữ VB.Net

```
Public Sub InsertionSort()
    Dim inner, outer, temp As Integer
    For outer = 1 To numElements - 1
        temp = arr(outer)
        inner = outer
        While (inner > 0 AndAlso (arr(inner - 1) >= temp))
            arr(inner) = arr(inner - 1)
            inner -= 1
        End While
        arr(inner) = temp
    Next
End Sub
```

- ánh giá giá trị thu được: phép tính giá trị thu được phụ thuộc vào số lần so sánh. Nếu tại thời điểm nào đó, tất cả các phần tử đã được sắp xếp thì vị trí chèn thích hợp. Do vậy số lần so sánh tối đa là:

$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$$

## ánh giá các gi i thu\_t s p x p

- C 3 GT u có ph c t p là  $O(n^2)$

## Bài toán tìm kiếm

- Tập dãy số  $a_1, a_2, \dots, a_N$  cần tìm
- Cần tìm phần tử có giá trị  $x$  xem có trong mảng không

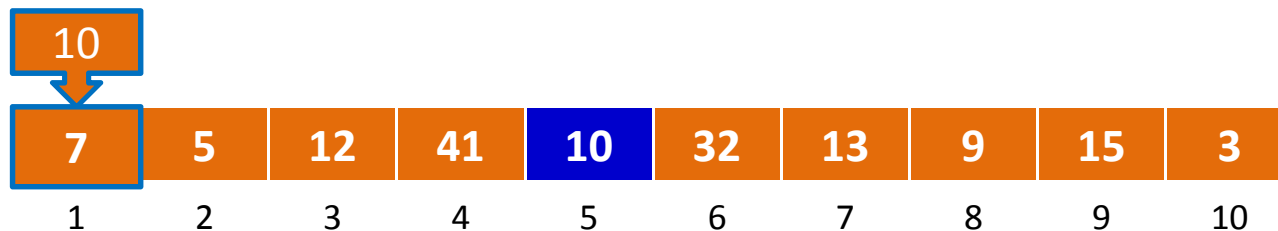


# Các gì i thu t tìm ki m

- Có 2 gì i thu t tìm ki m th ng áp d ng:
  - Tìm ki m tuy n tính: th ng th c hi n v i các m ng ch a c s p x p th t
  - Tìm ki m nh phân: th ng th c hi n v i các m ng ã s p x p th t

# Tìm kiếm tuyến tính

- Ý tưởng: Tiến hành so sánh  $x$  lần lượt với các phần tử liên tiếp, từ đầu đến cuối... của mảng cho đến khi gặp được phần tử có giá trị bằng  $x$  cần tìm, hoặc đã tìm đến hết mảng mà không thấy  $x$ .
- Ví dụ: Tìm phần tử có giá trị  $x = 10$



Đã tìm thấy tại vị trí 5

## Giới thiệu thuật tìm kiếm tuyến tính

- Đầu vào: mảng  $a[i]$  và  $x$
- Đầu ra: Trả về True nếu tìm thấy, ngược lại trả về False

Bước 1:

$i = 1;$  // bắt đầu phần tử đầu tiên của dãy

Bước 2:

So sánh  $a[i]$  với  $x$ , có 2 khả năng:

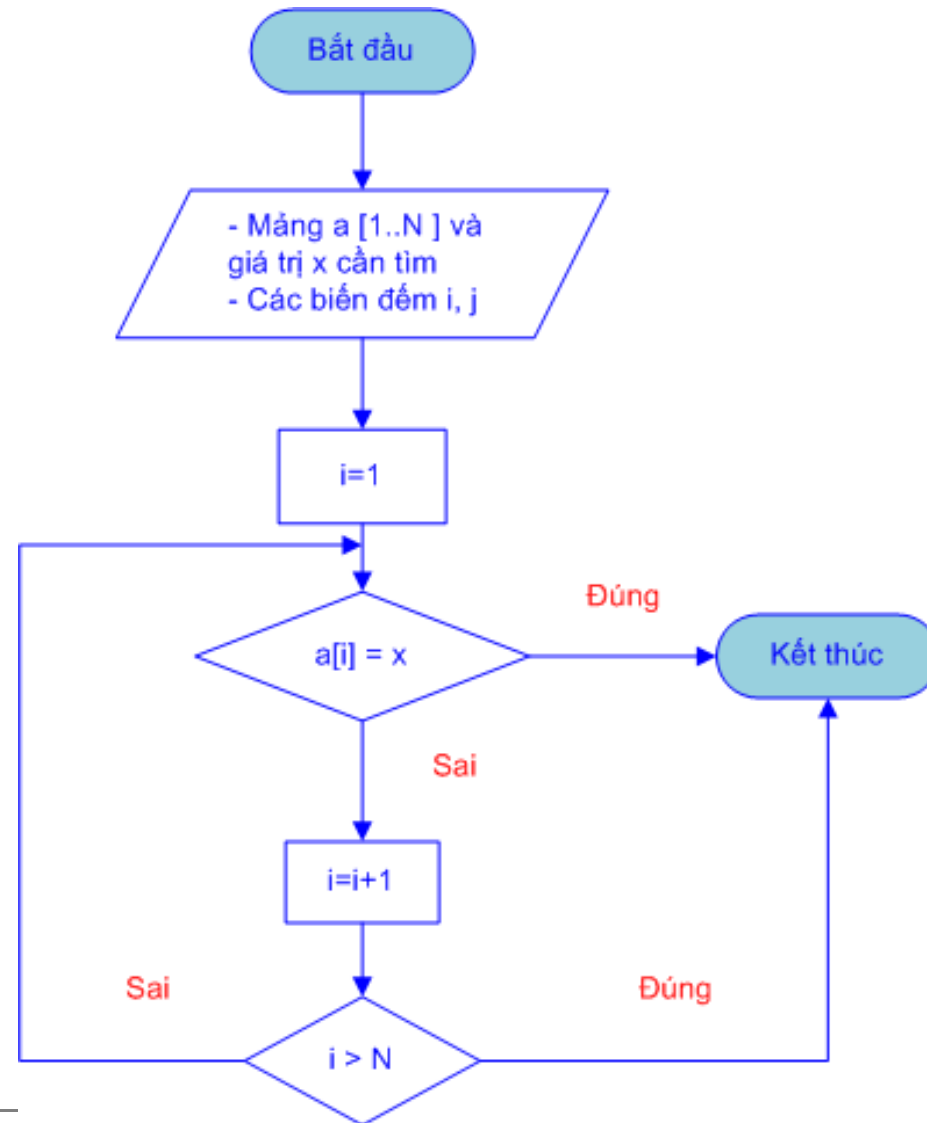
- $a[i] = x$ : Tìm thấy. Dừng
- $a[i] \neq x$ : Sang Bước 3.

Bước 3:

- $i = i + 1;$  // xét tiếp phần tử kế trong mảng
- Nếu  $i > N$ : Hết mảng, không tìm thấy. Dừng  
Ngược lại: Lặp lại Bước 2.



# Lưu ý thuật tìm kiếm tuyến tính



# Cài đặt giải thuật

```

Function SeqSearch(ByVal arr() As Integer,
ByVal sValue As Integer) As Integer
    Dim index As Integer
    For index = 0 To arr.GetUpperBound(0)
        If (arr(index) = sValue) Then
            Return True
        End If
    Next
    Return False
End Function

```

# Cài đặt giải thuật

Giải thuật trả về vị trí của phần tử tìm thấy trong mảng, hoặc trả về -1 nếu không tìm thấy:

```
Function SeqSearch(ByVal arr() As Integer,
    ByVal sValue As Integer) As Integer
    Dim index As Integer
    For index = 0 To arr.GetUpperBound(0)
        If (arr(index) = sValue) Then
            Return index
        End If
    Next
    Return -1
End Function
```

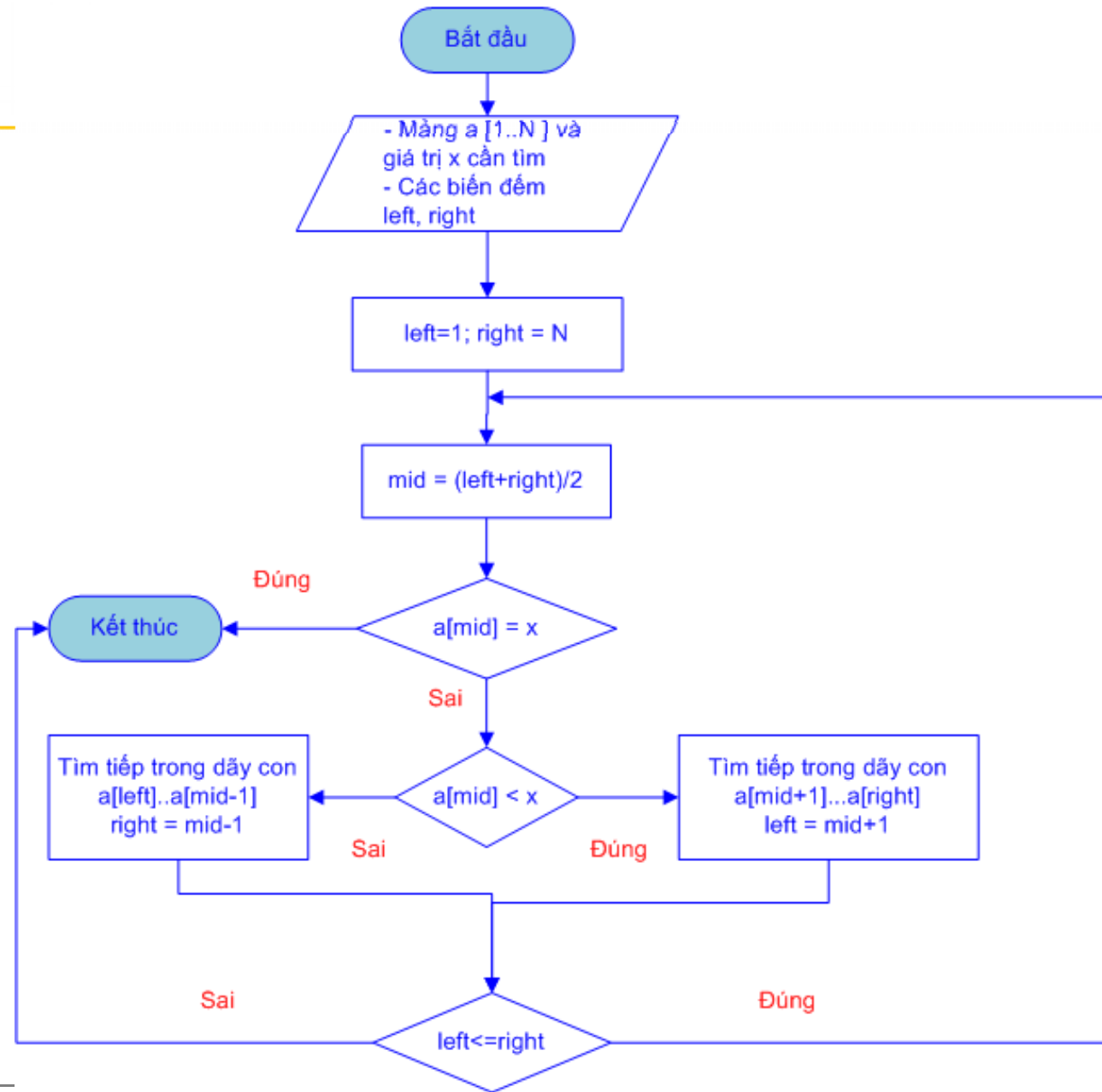
# Tìm kiếm nhị phân

Ý tưởng:

- Tìm kiếm kiểu “tra tì n”
- Giả i thu t tìm cách giả i h n ph m vi tìm ki m sau m i l n so sánh **x** v i m t ph n t trong dãy ã c s p x p.
- T i m i b c, so sánh **x** v i **ph n t n m** v trí gi a **c a dãy** tìm ki m hi n hành:
  - N u x nh h n thì s tìm ki m n a tr c c a dãy.
  - Ng c l i, tìm n a sau c a dãy.

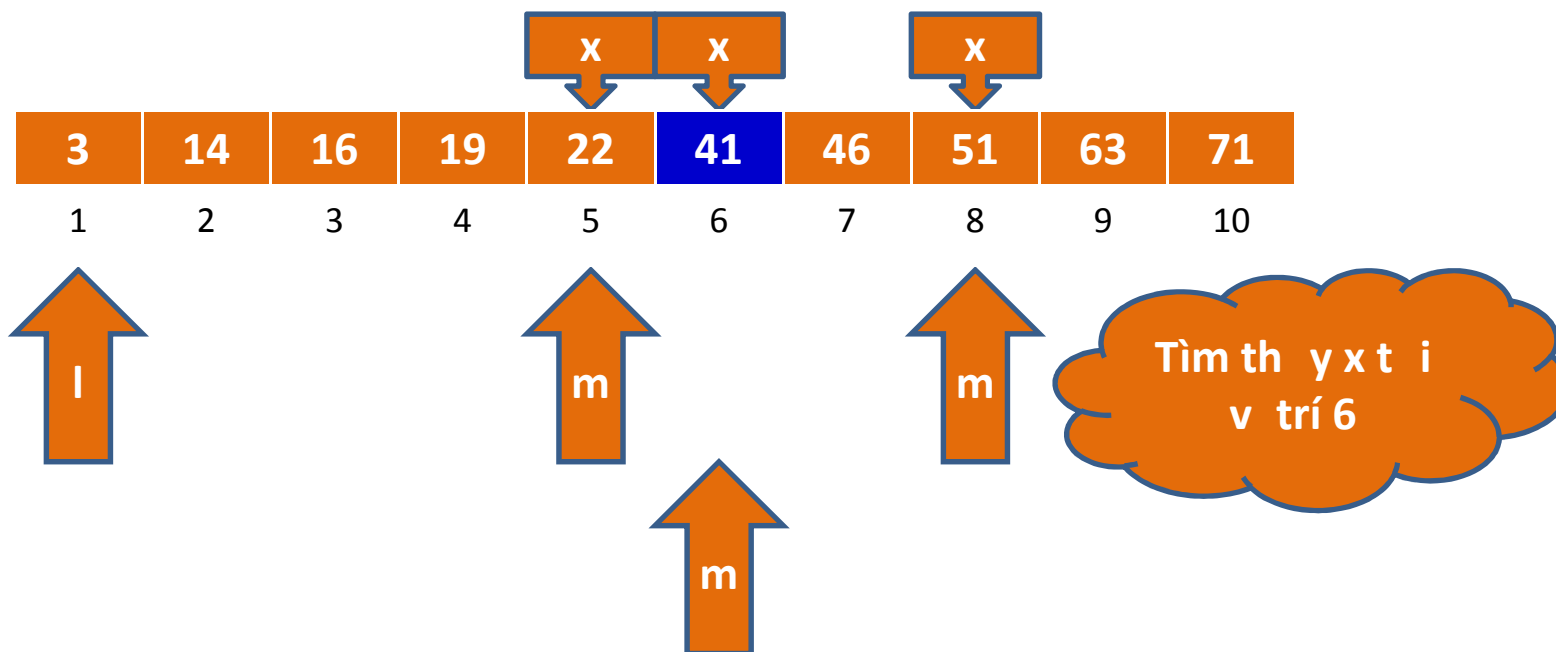
- Bước 1:  $\text{left} = 1; \text{right} = N;$  // tìm kiếm trên tất cả các phần tử
- Bước 2:
  - $\text{mid} = (\text{left} + \text{right}) / 2;$  // lấy mức so sánh
  - So sánh  $a[\text{mid}]$  với  $x$ , có 3 khả năng:
    - ❖  $a[\text{mid}] = x$ : Tìm thấy. Dừng
    - ❖  $a[\text{mid}] > x$ : // tìm tiếp  $x$  trong dãy con  $a_{\text{left}} \dots a_{\text{mid}-1}$
    - ❖  $\text{right} = \text{mid} - 1;$
    - ❖  $a[\text{mid}] < x$ : // tìm tiếp  $x$  trong dãy con  $a_{\text{mid}+1} \dots a_{\text{right}}$
    - $\text{left} = \text{mid} + 1;$
- Bước 3:
  - Nếu  $\text{left} \leq \text{right}$  // còn phần tử chưa xét tìm tiếp.
  - Lặp lại Bước 2.
  - Ngừng lại: Dừng // Đã xét hết tất cả các phần tử.

# L U G I I T H U T



# Tìm kiếm nhị phân

Ví dụ minh họa tìm  $x = 41$



## ánh giá các giải thuật tìm kiếm

- Nếu tìm kiếm trên mảng liên tiếp và các số sắp xếp thì sử dụng phương pháp tìm kiếm tuyến tính
- Nếu tìm kiếm trên mảng đã sắp xếp thì sử dụng phương pháp tìm kiếm nhị phân.
- Phân tích độ phức tạp toán học của hai phương pháp:  $O(n)$



- Sắp xếp và tìm kiếm là những bài toán quan trọng trong lập trình máy tính.
- Việc sắp xếp và tìm kiếm thường được tiến hành trên mảng
- 3 giai đoạn thực hiện sắp xếp và tìm kiếm phức tạp  $O(n^2)$
- 2 giai đoạn tìm kiếm và tìm kiếm phức tạp  $O(n)$ :
  - Tìm kiếm tuyến tính: thực hiện trên mảng chưa được sắp xếp
  - Tìm kiếm nhị phân: thực hiện trên mảng đã được sắp xếp