

CHƯƠNG 4

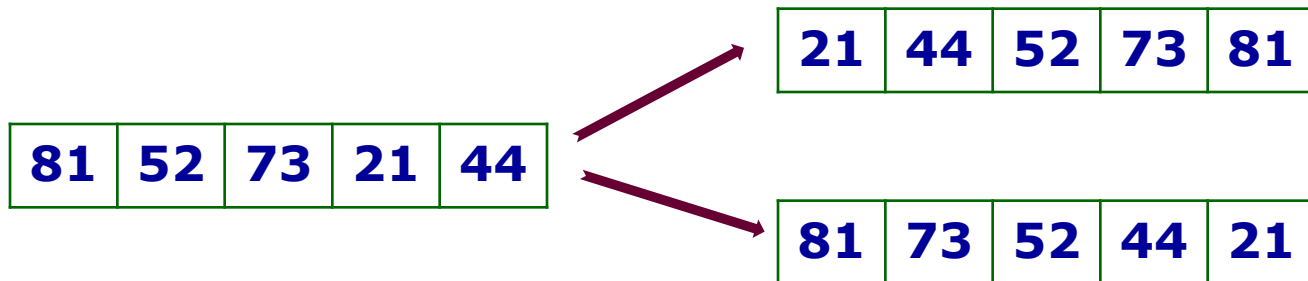
CÁC THUẬT TOÁN SẮP XẾP

Nội dung

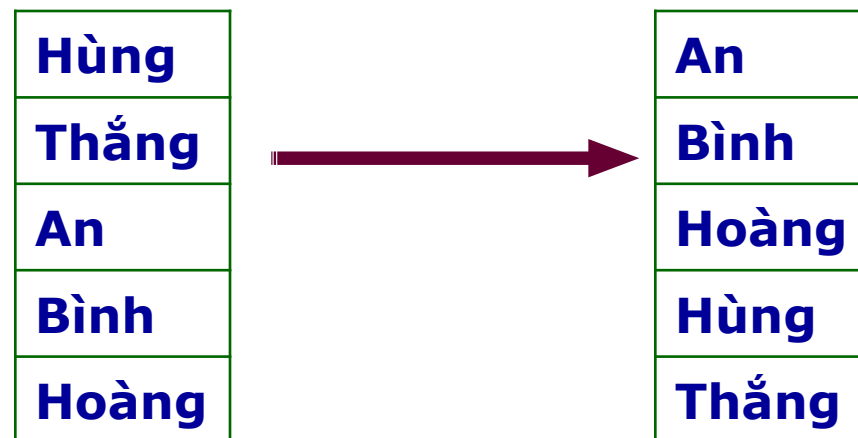
1. Khái niệm sắp xếp
2. Phát biểu bài toán sắp xếp
3. Sắp xếp trong và sắp xếp ngoài
4. Các phương pháp sắp xếp đơn giản
 - ✓ Sắp xếp nổi bọt – bubble sort
 - ✓ Sắp xếp lựa chọn – selection sort
 - ✓ Sắp xếp chèn – insertion sort
5. Sắp xếp nâng cao
 - ✓ Sắp xếp phân đoạn – quick sort
 - ✓ Sắp xếp vun đống – heap sort
 - ✓ Sắp xếp trộn – merge sort

Khái niệm sắp xếp

- Vấn đề sắp xếp
 - Sắp xếp dãy số



- Sắp xếp danh sách học sinh



Khái niệm sắp xếp

- **Khái niệm**

- Sắp xếp là việc biến đổi vị trí của một tập đối tượng theo một trật tự mới nhằm thỏa mãn một mục đích.

- **Mục đích**

- Giúp việc tìm kiếm được thực hiện nhanh hơn.
- Sắp thứ tự để in bảng biểu.

Bài toán sắp xếp



Đầu vào:

Dãy n đối tượng, mỗi đối tượng có một khóa sắp xếp.

Đầu ra:

Dãy n đối tượng được sắp xếp theo trật tự của khóa.

Bài toán sắp xếp



Đầu vào:

$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$
5	-1	7	3	-4

Đầu ra:

$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$
-4	-1	3	5	7

Khái niệm sắp xếp trong và sắp xếp ngoài

- **Sắp xếp trong:**

- Dãy đối tượng được sắp có mặt đầy đủ ở bộ nhớ trong (RAM).
- Sắp xếp danh sách được lưu trong mảng một chiều: danh sách các số, danh sách học sinh, ...
- Sắp tại chỗ: Đổi vị trí các phần tử trong mảng.

- **Sắp xếp ngoài:**

- Dãy đối tượng được sắp chưa có mặt đầy đủ ở bộ nhớ trong.
- Sắp xếp dữ liệu được lưu trong tệp.

Ba phương pháp sắp xếp đơn giản

1. Sắp xếp nổi bọt – Bubble sort
2. Sắp xếp lựa chọn – Selection sort
3. Sắp xếp chèn – Insertion sort

4.1. Sắp xếp nổi bọt – Bubble sort

- Ý tưởng giải thuật

- Thực hiện việc đổi chỗ các cặp phần tử kế tiếp trong dãy đối tượng cần sắp nếu chúng chưa đúng thứ tự.

- Ví dụ minh họa:

- Cho dãy số có 5 số nguyên ($n = 5$) được lưu trong mảng x như dưới đây.

- Dãy x :

$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$
5	-1	7	3	-4

- Yêu cầu: Sắp xếp dãy số theo chiều tăng dần.

4.1. Sắp xếp nổi bọt – Bubble sort

- Thực hiện:

- Duyệt lần 1: So sánh các cặp phần tử kế tiếp của dãy là: $(x[0], x[1])$; $(x[1], x[2])$; $(x[2], x[3])$; $(x[3], x[4])$, nếu cặp nào trái thứ tự thì đổi chỗ.

$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$
5	-1	7	3	-4
-1	5	7	3	-4
-1	5	7	3	-4
-1	5	3	7	-4
-1	5	3	-4	7

4.1. Sắp xếp nổi bọt – Bubble sort

- **Nhận xét:**

- Sau lần duyệt vừa rồi ta thấy phần tử lớn nhất được chuyển về cuối dãy, nghĩa là nó đứng đúng vị trí.
- Các phần tử còn lại vẫn chưa đúng thứ tự.
- Duyệt lần 2:

x[0]	x[1]	x[2]	x[3]	x[4]
-1	5	3	-4	7
-1	5	3	-4	7
-1	3	5	-4	7
-1	3	-4	5	7

4.1. Sắp xếp nổi bọt – Bubble sort

- Duyệt lần 3:

x[0]	x[1]	x[2]	x[3]	x[4]
-1	3	-4	5	7
-1	3	-4	5	7
-1	-4	3	5	7

- Duyệt lần 4:

x[0]	x[1]	x[2]	x[3]	x[4]
-1	-4	3	5	7
-4	-1	3	5	7

4.1. Sắp xếp nổi bọt – Bubble sort

- Sau 4 lần duyệt dãy, so sánh và đổi chỗ các cặp phần tử kế tiếp nhau trong dãy ta được dãy sắp xếp theo chiều tăng dần.

x[0]	x[1]	x[2]	x[3]	x[4]
-4	-1	3	5	7

4.1. Sắp xếp nổi bọt – Bubble sort

- Ví dụ 2:

- Cho dãy số sau

x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]
-35	41	23	15	-62	12	71

- Yêu cầu:

- ✓ Minh họa quá trình sắp xếp dãy theo chiều giảm dần dựa vào ý tưởng trên.

4.1. Sắp xếp nổi bọt – Bubble sort

- **Thiết kế giải thuật**

- Thực hiện $n-1$ lần duyệt dãy (từ trái sang phải):

Sử dụng vòng lặp **for** ($i = 1; i \leq n-1; i++$)

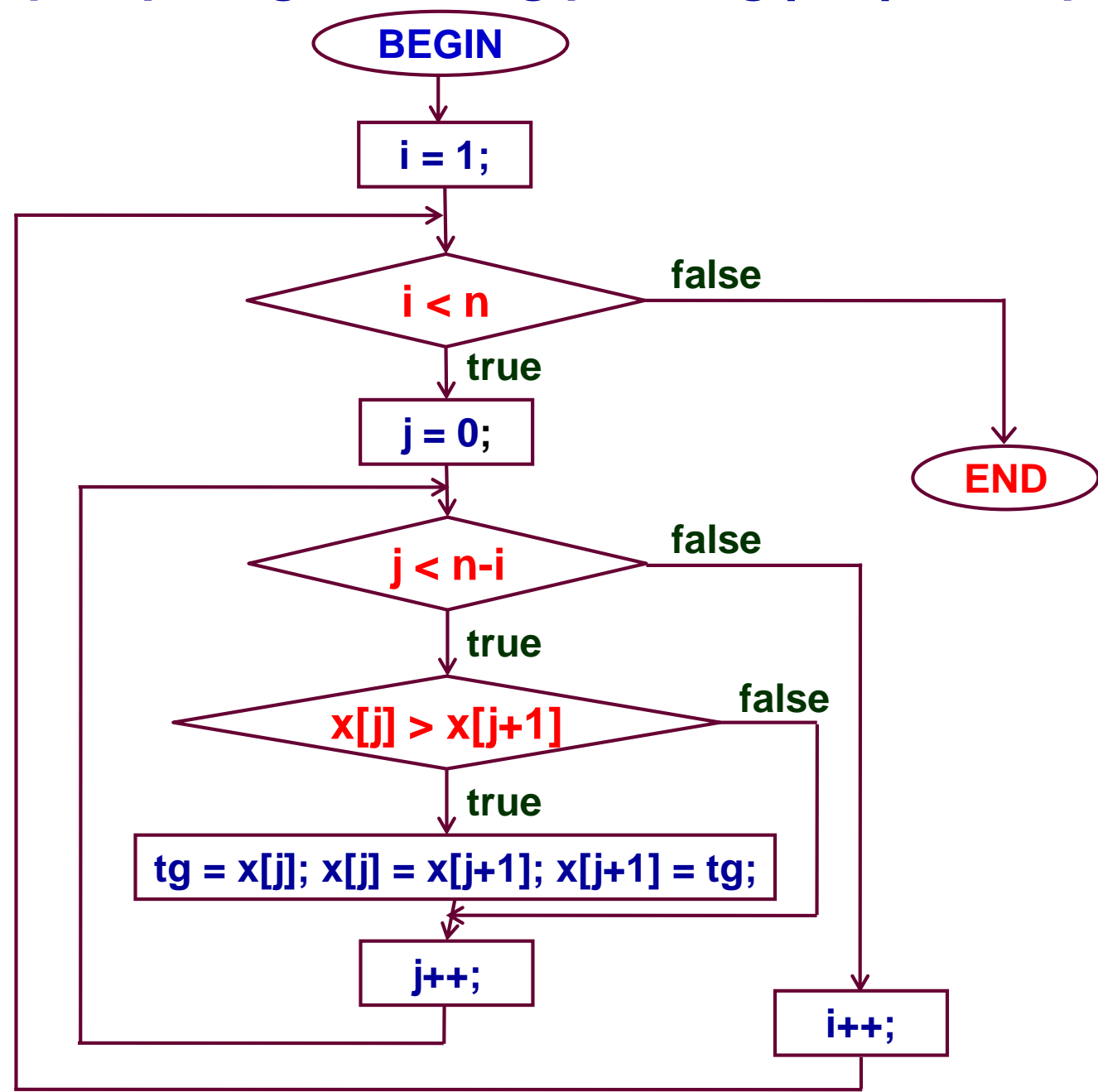
- Mỗi lần duyệt, so sánh các cặp phần tử kế tiếp nhau là $x[j]$ và $x[j+1]$, nếu trái chiều thì đổi chỗ $x[j]$ và $x[j+1]$ với ($j=0 \rightarrow n-i-1, i=1 \rightarrow n-1$):

Duyệt: Sử dụng vòng lặp **for** ($j = 0; j < n-i; j++$)

So sánh và đổi chỗ: Sử dụng cấu trúc rẽ nhánh

```
if ( $x[j] > x[j+1]$ ) {  
     $tg = x[j]; x[j] = x[j+1]; x[j+1] = tg;$   
}
```

■ Lưu đồ giải thuật sắp tăng dần bằng phương pháp nổi bọt



4.1. Sắp xếp nổi bọt – Bubble sort

- Giải thuật: Sắp xếp dãy n số nguyên theo chiều tăng dần.

```
void bubbleSort(int x[], int n)
{
    for (int i=1; i<=n-1; i++)
        for (int j=0; j<n-i; j++)
            if (x[j]>x[j+1])
            {
                int tg = x[j];
                x[j] = x[j+1];
                x[j+1] = tg;
            }
}
```

4.1. Sắp xếp nổi bọt – Bubble sort

- Cài đặt ứng dụng thực hiện các yêu cầu:
 - Khởi tạo một số nguyên dương n .
 - Khởi tạo dãy có n số nguyên.
 - Sắp xếp dãy theo chiều tăng dần bằng giải thuật sắp xếp nổi bọt.
 - In dãy vừa sắp ra màn hình.

4.1. Sắp xếp nổi bọt – Bubble sort

- **Bài tập 1:** Cho dãy số nguyên

34 14 24 54 84 64 94 74 04

- Minh họa việc sắp xếp dãy số theo chiều tăng dần (giảm dần) bằng phương pháp nổi bọt.
- Cài đặt chương trình sắp xếp dãy số.

- **Bài tập 2:** Cho dãy từ

John Wenger Anna Henry Thor Terry Ozil Adam Dennis

- Minh họa việc sắp xếp dãy từ theo trật tự từ điển (ngược lại với trật tự từ điển) bằng phương pháp nổi bọt.
- Cài đặt chương trình sắp xếp dãy từ.

4.1. Sắp xếp nổi bọt – Bubble sort

- **Bài tập 3: Cài đặt chương trình thực hiện các yêu cầu:**
 - Nhập vào một danh sách học sinh ($0 < n < 100$, n nhập từ bàn phím), mỗi học sinh gồm các thông tin: Mã học sinh, họ và tên, năm sinh và điểm trung bình.
 - Sắp xếp danh sách theo chiều tăng dần của tên học sinh bằng giải thuật sắp xếp nổi bọt.
 - In danh sách vừa sắp ra màn hình.
 - Sắp xếp danh sách theo chiều giảm dần của điểm trung bình bằng thuật sắp xếp nổi bọt.
 - In danh sách ra màn hình.

Ví dụ cách trình bày bài toán

$i = 0$	j	0	1	2	3	4	5	6	7
	0	5	3	1	9	8	2	4	7
	1	3	5	1	9	8	2	4	7
	2	3	1	5	9	8	2	4	7
	3	3	1	5	9	8	2	4	7
	4	3	1	5	8	9	2	4	7
	5	3	1	5	8	2	9	4	7
	6	3	1	5	8	2	4	9	7
$i = 1$	0	3	1	5	8	2	4	7	9
	1	1	3	5	8	2	4	7	
	2	1	3	5	8	2	4	7	
	3	1	3	5	8	2	4	7	
	4	1	3	5	2	8	4	7	
	5	1	3	5	2	4	8	7	
$i = 2$	0	1	3	5	2	4	7	8	
	1	1	3	5	2	4	7		
	2	1	3	5	2	4	7		
	3	1	3	2	5	4	7		
	4	1	3	2	4	5	7		
$i = 3$	0	1	3	2	4	5	7		
	1	1	3	2	4	5			
	2	1	2	3	4	5			
	3	1	2	3	4	5			
$i = 4$	0	1	2	3	4	5			
	1	1	2	3	4				
	2	1	2	3	4				
$i = 5$	0	1	2	3	4				
	1	1	2	3					
$i = 6$	0	1	2	3					
		1	2						

4.2. Sắp xếp lựa chọn – Selection sort

- Ý tưởng giải thuật

- Dựa vào thuật toán tìm phần tử nhỏ nhất.
- Duyệt và tìm phần tử nhỏ nhất của dãy và đổi chỗ phần tử đó cho phần tử đầu dãy.

- Ví dụ

- Cho dãy số có 5 số nguyên ($n = 5$) được lưu trong mảng x như dưới đây.

- Dãy x :

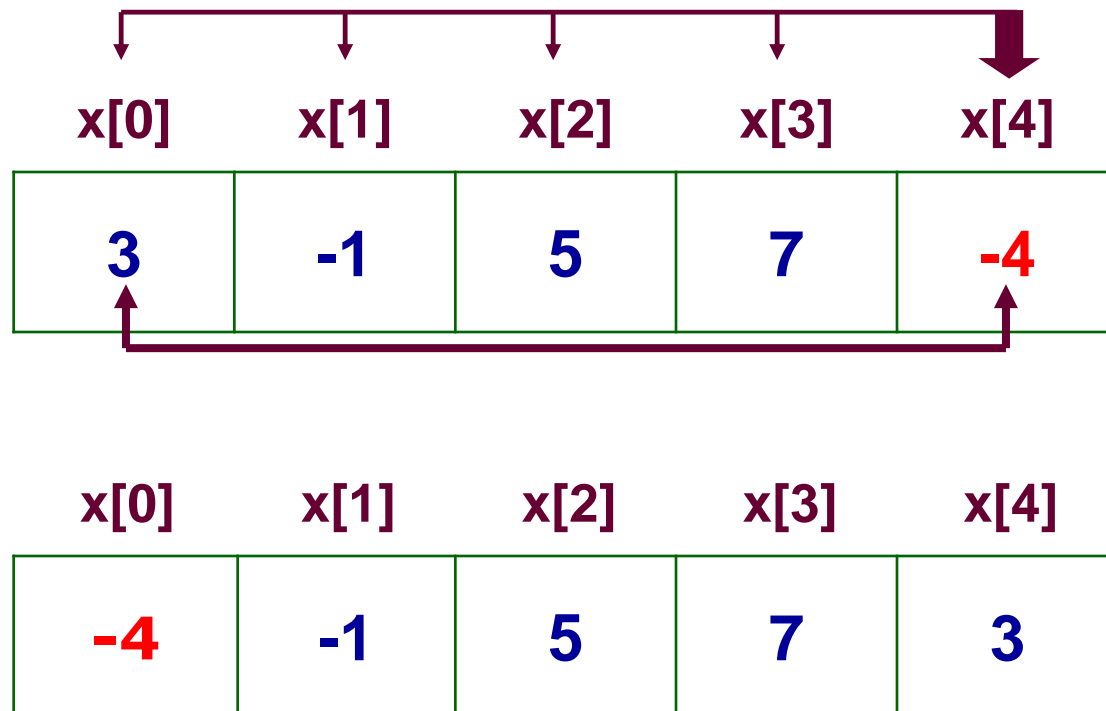
$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$
3	-1	7	5	-4

- Yêu cầu: Sắp xếp dãy số theo chiều tăng dần.

4.2. Sắp xếp lựa chọn – Selection sort

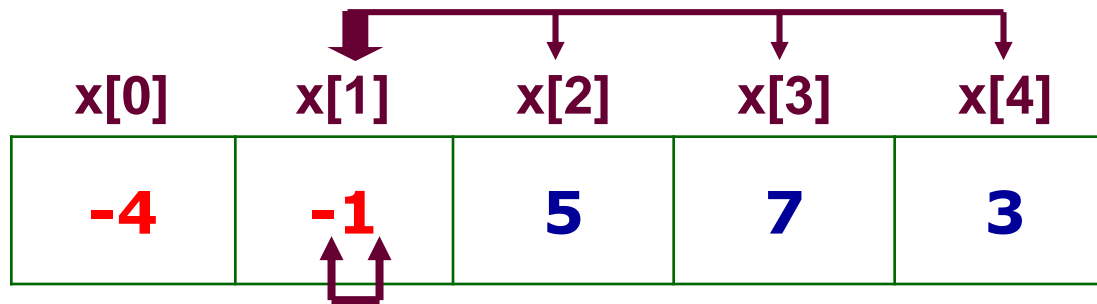
- Thực hiện:

- Duyệt lần 1: Tìm phần tử có giá trị nhỏ nhất trong số các phần tử từ $x[0]$ đến $x[4]$, được $x[4] = -4$, đổi chỗ $x[4]$ với $x[0]$.



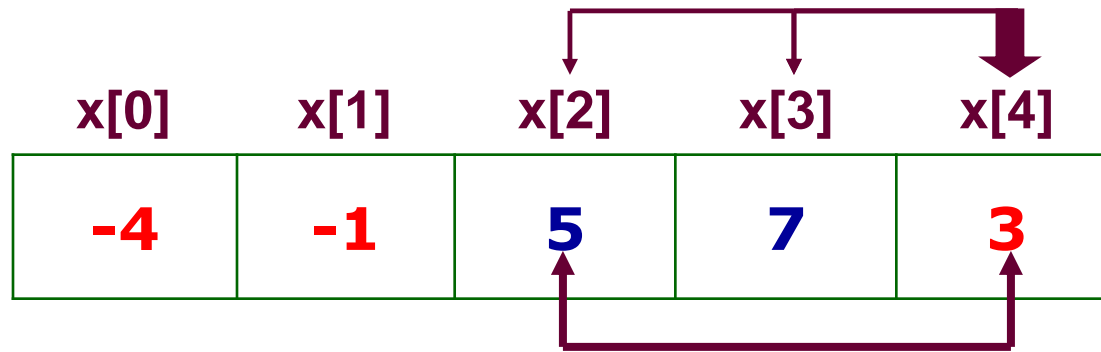
4.2. Sắp xếp lựa chọn – Selection sort

- **Duyệt lần 2:** Tìm phần tử có giá trị nhỏ nhất trong số các phần tử từ $x[1]$ đến $x[4]$, được $x[1] = -1$, đổi chỗ $x[1]$ với $x[1]$.



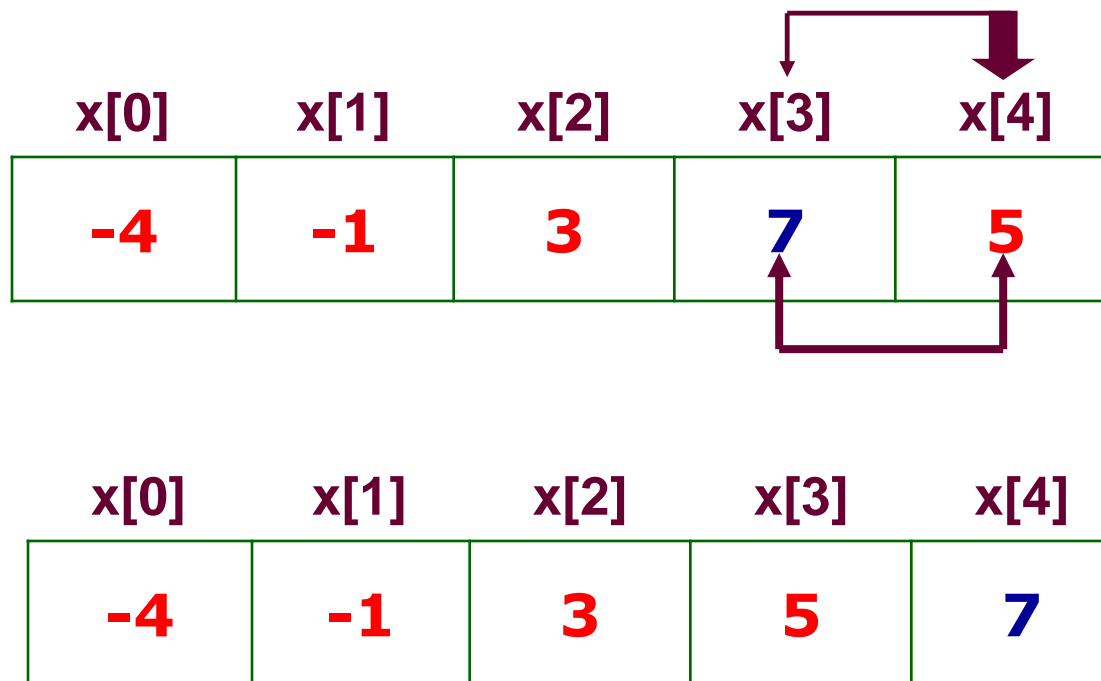
4.2. Sắp xếp lựa chọn – Selection sort

- Duyệt lần 3:** Tìm phần tử có giá trị nhỏ nhất trong số các phần tử từ $x[2]$ đến $x[4]$, được $x[4] = 3$, đổi chỗ $x[2]$ với $x[4]$.



4.2. Sắp xếp lựa chọn – Selection sort

- Duyệt lần 4:** Tìm phần tử có giá trị nhỏ nhất trong số các phần tử từ $x[3]$ đến $x[4]$, được $x[4] = 5$, đổi chỗ $x[3]$ với $x[4]$.



4.2. Sắp xếp lựa chọn – Selection sort

- Ví dụ áp dụng:
 - Cho dãy số nguyên như sau:

x[0]	x[1]	x[2]	x[3]	x[4]	x[5]
53	-21	67	15	82	-14

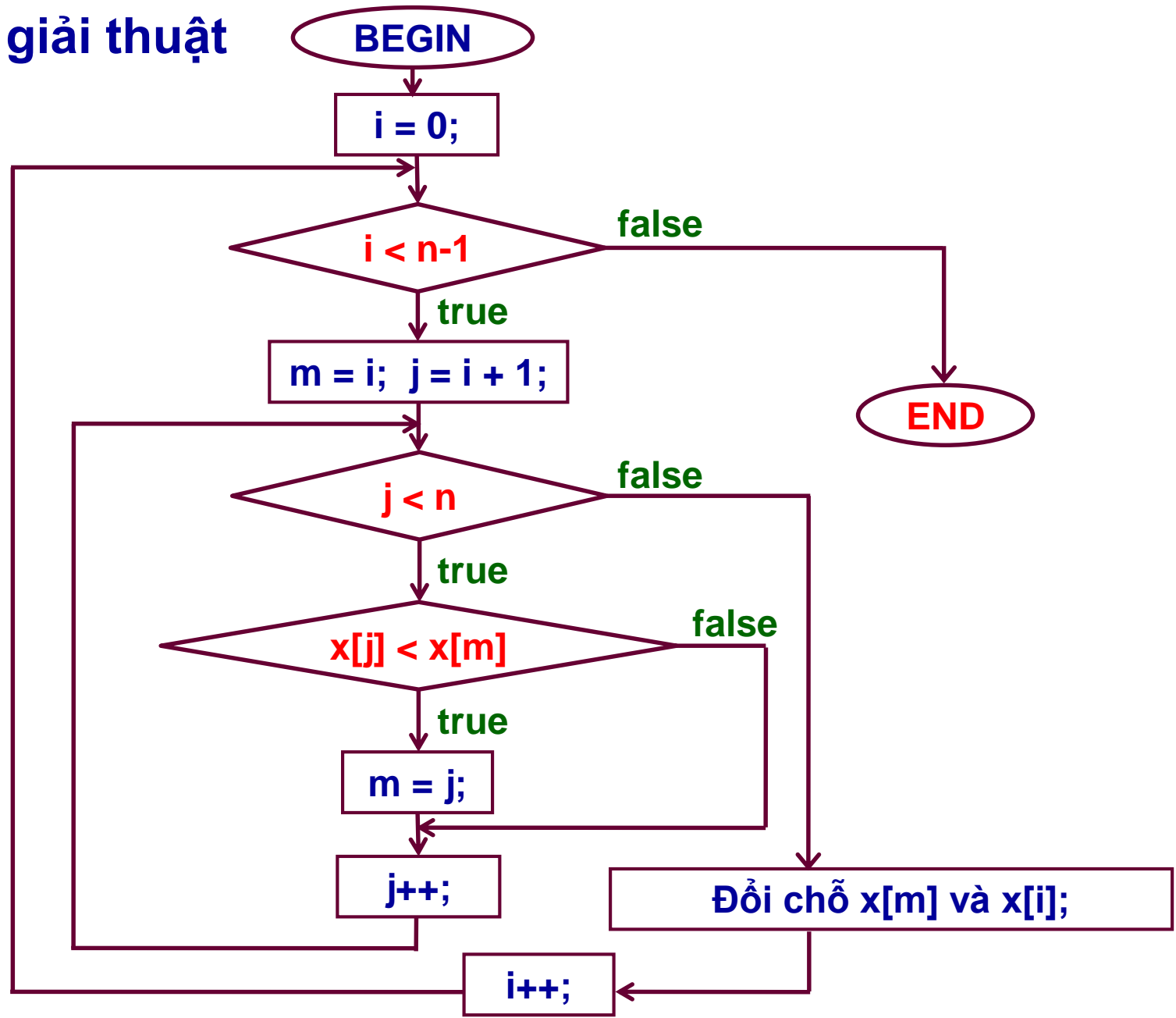
- Yêu cầu:
 - Dựa vào ý tưởng trên, minh họa việc sắp xếp dãy số theo chiều giảm dần.

4.2. Sắp xếp lựa chọn – Selection sort

- **Thiết kế giải thuật**

- Thực hiện $n - 1$ lần duyệt dãy (từ trái sang phải).
- Mỗi lần duyệt tìm phần tử nhỏ nhất trong số các phần tử từ $x[i]$ đến $x[n-1]$, giả sử là $x[m]$, đổi chỗ $x[m]$ và $x[i]$ (với $i = [0, n-2]$)

■ Lưu đồ giải thuật



4.2. Sắp xếp lựa chọn – Selection sort

- Giải thuật

```
void selectionSort(int X[], int n) {  
    for (int i=0; i<n-1; i++) {  
        int m=i;  
        for (int j=i+1; j<n; j++)  
            if (X[j]<X[m])  
                m = j;  
        int tg = X[m];  
        X[m] = X[i];  
        X[i] = tg;  
    }  
}
```

4.2. Sắp xếp lựa chọn – Selection sort

- Cài đặt ứng dụng thực hiện các yêu cầu:
 - Khởi tạo một số nguyên dương n .
 - Khởi tạo dãy có n số nguyên.
 - Sắp xếp dãy theo chiều tăng dần bằng giải thuật sắp xếp lựa chọn.
 - In dãy vừa sắp ra màn hình.

4.2. Sắp xếp lựa chọn – Selection sort

- **Bài tập 1:** Cho dãy số nguyên

34 14 24 54 84 64 94 74 04

- Minh họa việc sắp xếp dãy số theo chiều tăng dần (giảm dần) bằng phương pháp sắp xếp lựa chọn.
- Cài đặt chương trình sắp xếp dãy số.

- **Bài tập 2:** Cho dãy từ

John Wenger Anna Henry Thor Terry Ozil Adam Dennis

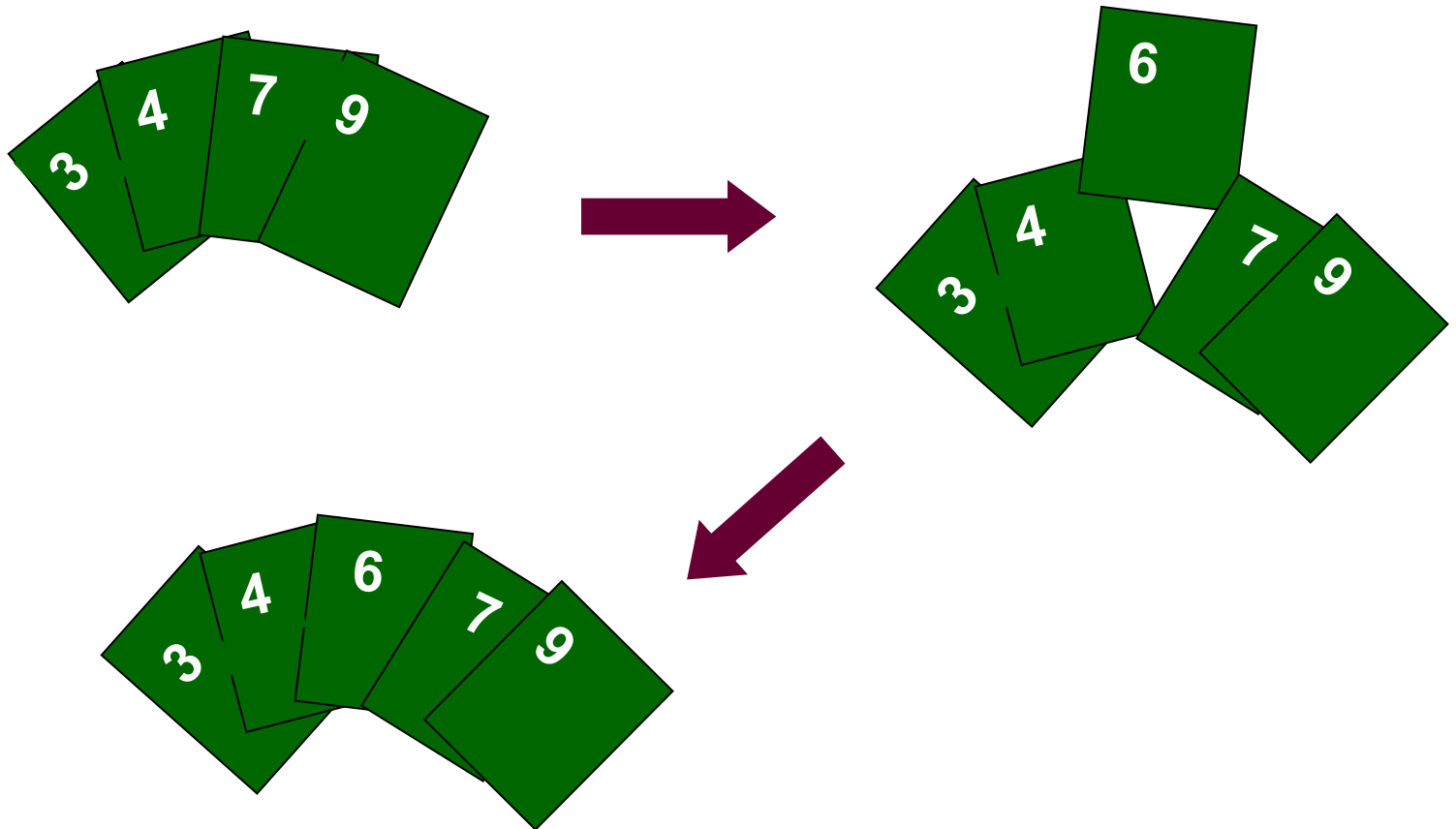
- Minh họa việc sắp xếp dãy từ theo trật tự từ điển (ngược lại với trật tự từ điển) bằng phương pháp sắp xếp lựa chọn.
- Cài đặt chương trình sắp xếp dãy từ.

4.2. Sắp xếp lựa chọn – Selection sort

- **Bài tập 3: Cài đặt chương trình thực hiện các yêu cầu:**
 - Nhập vào một danh sách học sinh ($0 < n < 100$, n nhập từ bàn phím), mỗi học sinh gồm các thông tin: Mã học sinh, họ và tên, năm sinh và điểm trung bình.
 - Sắp xếp danh sách theo chiều tăng dần của tên học sinh bằng giải thuật sắp xếp lựa chọn.
 - In danh sách vừa sắp ra màn hình.
 - Sắp xếp danh sách theo chiều giảm dần của điểm trung bình bằng thuật sắp xếp lựa chọn.
 - In danh sách ra màn hình.

4.3. Sắp xếp chèn – Insertion sort

- Bài toán gợi ý tưởng:



4.3. Sắp xếp chèn – Insertion sort

- Ý tưởng giải thuật

- Chèn một phần tử vào dãy đã được sắp xếp sao cho trật tự của dãy không bị thay đổi..
- Duyệt và tìm vị trí thích hợp cho phần tử mới trong dãy.
- Chèn phần tử mới vào vị trí thích hợp.

- Ví dụ

- Cho dãy số có 5 số nguyên ($n = 5$) được lưu trong mảng x như dưới đây.

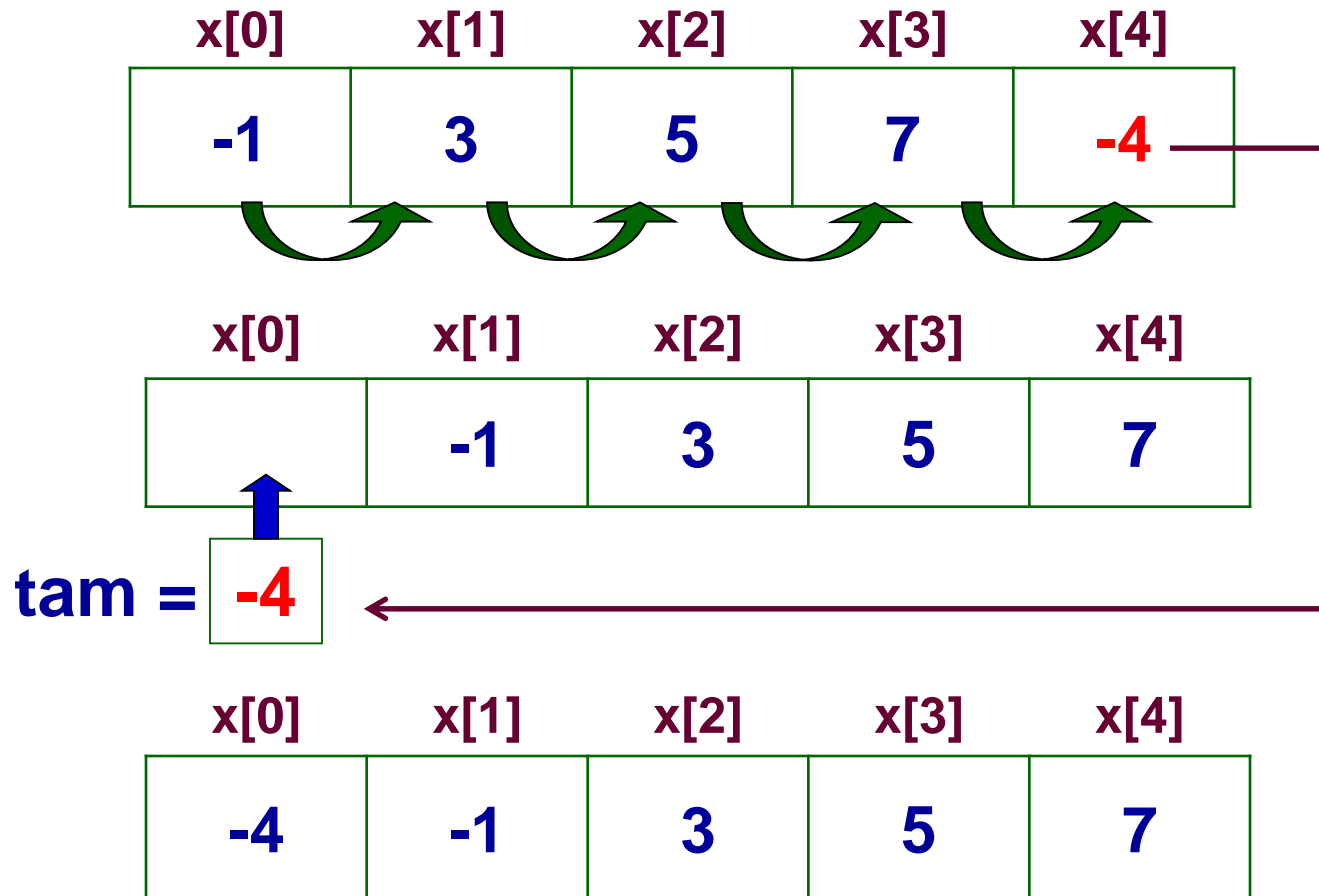
- Dãy x :

$x[0]$	$x[1]$	$x[2]$	$x[3]$	$x[4]$
3	-1	7	5	-4

- Yêu cầu: Sắp xếp dãy số theo chiều tăng dần.

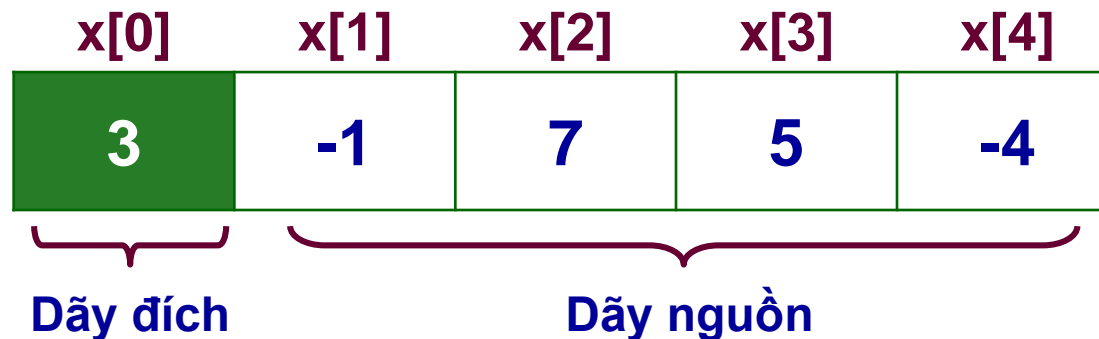
4.3. Sắp xếp chèn – Insertion sort

- **Thể hiện ý tưởng:** Chèn số **-4** ở cuối dãy vào phía trước đã sắp xếp theo chiều tăng dần.



4.3. Sắp xếp chèn – Insertion sort

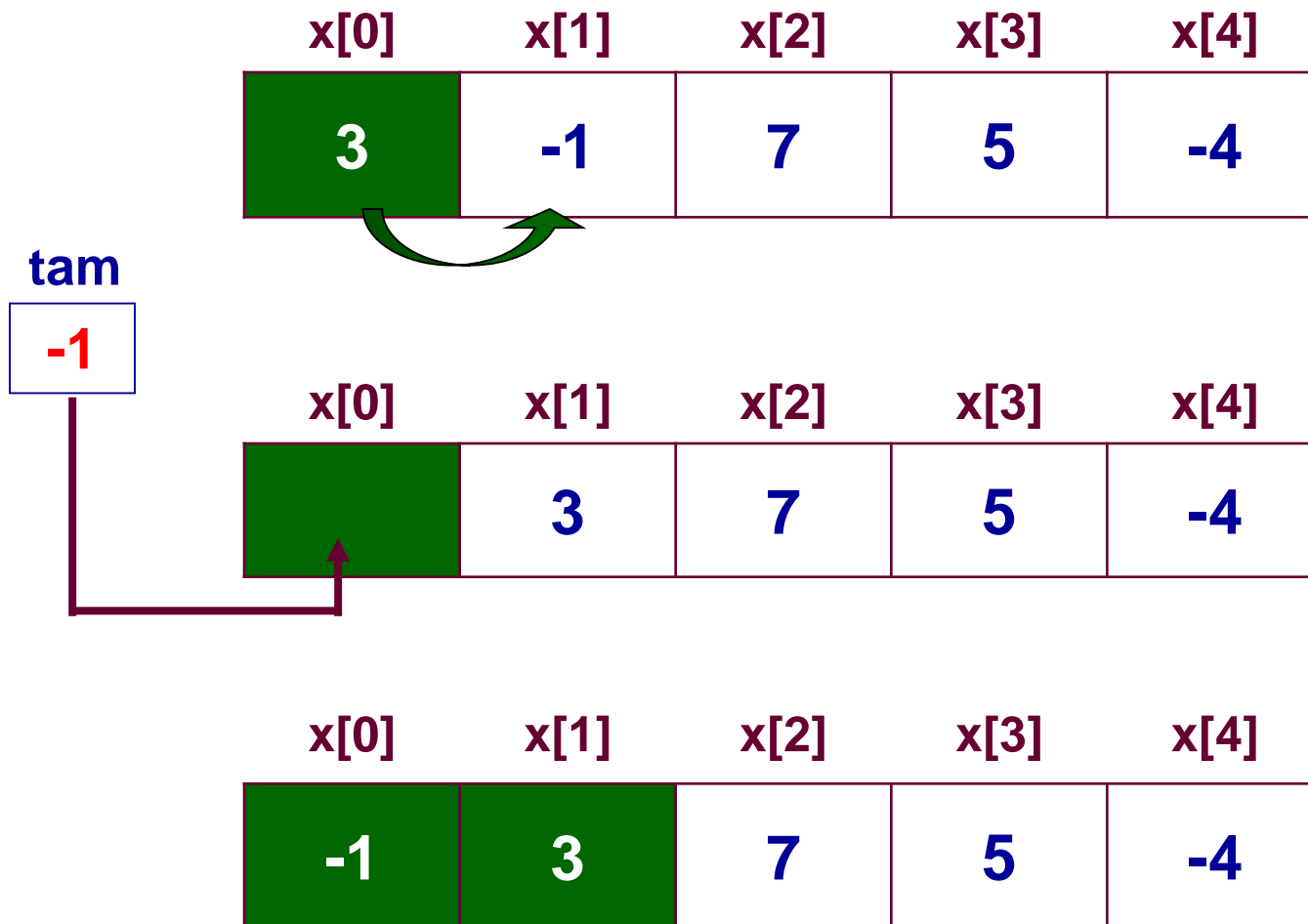
- **Đưa ý tưởng vào bài toán sắp xếp:**
 - Xem dãy cần sắp gồm 2 dãy nối tiếp.
 - Dãy trái (dãy đích) gồm các phần tử được sắp, dãy phải (dãy nguồn) là các phần tử chưa được sắp.



- Lấy phần tử đầu dãy nguồn chèn vào vị trí thích hợp trong dãy đích.

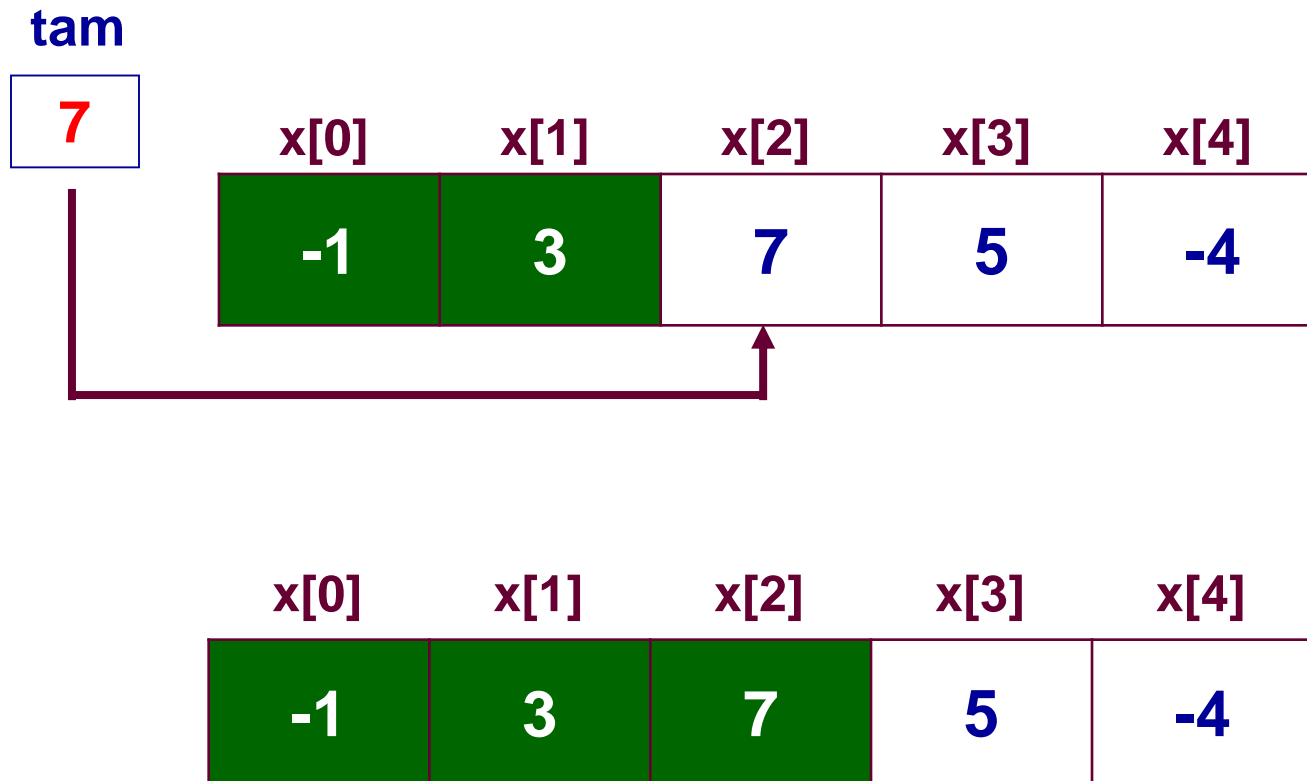
4.3. Sắp xếp chèn – Insertion sort

- Lượt chia 1: Chia ở vị trí $x[1]$, $\text{tam} = x[1] = -1$.



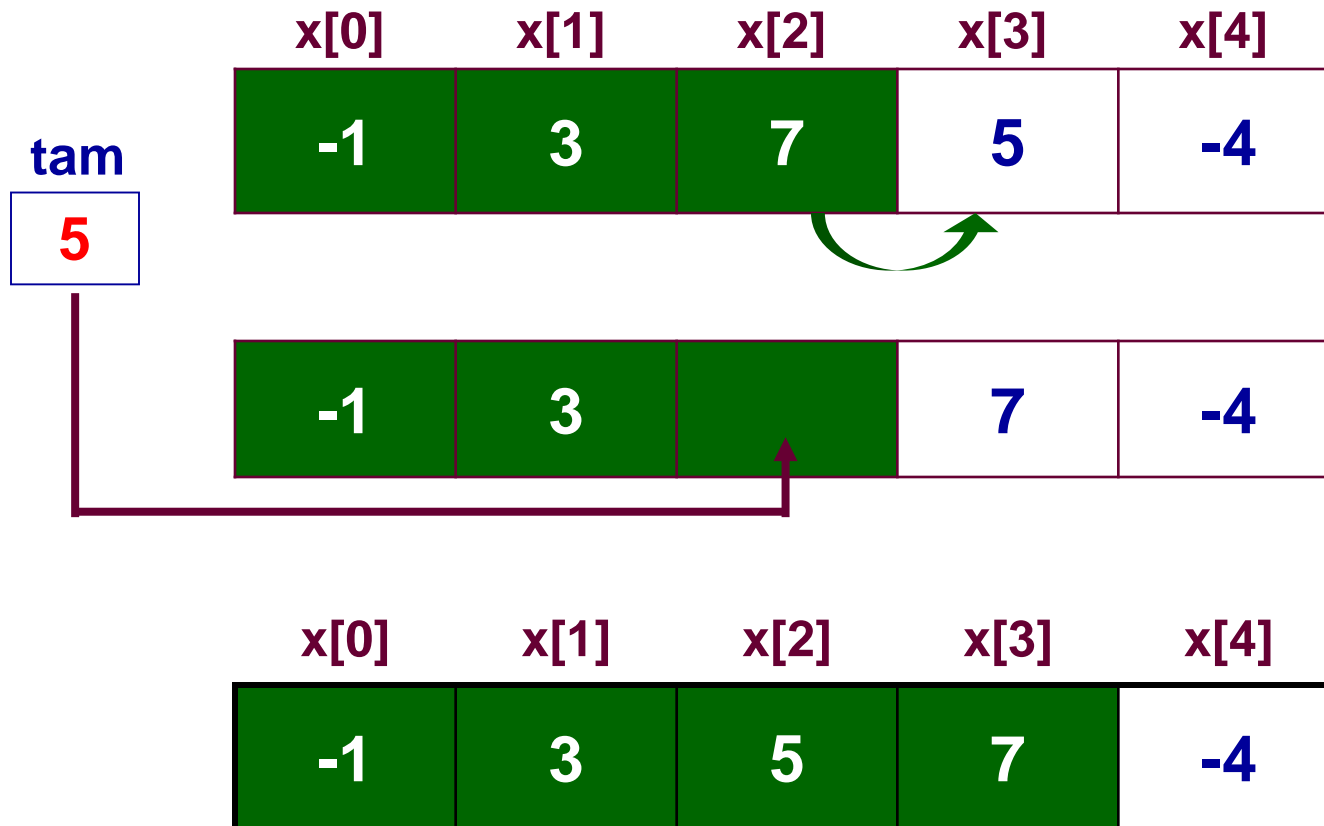
4.3. Sắp xếp chèn – Insertion sort

- **Lượt chia 2:** Chia ở vị trí $x[2]$, $\text{tam} = x[2] = 7$;



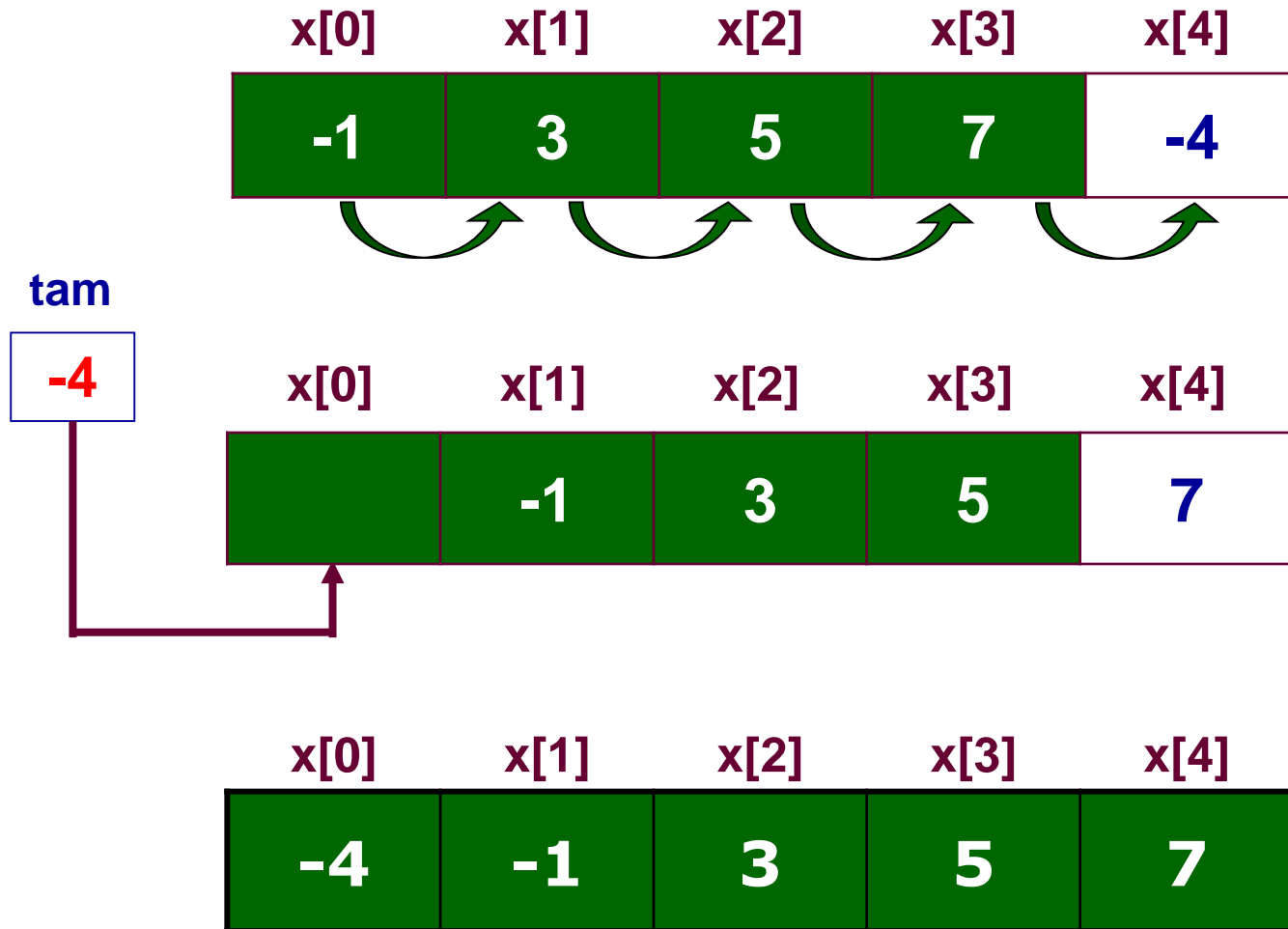
4.3. Sắp xếp chèn – Insertion sort

- Lượt chia 3: Chia ở vị trí $x[3]$, $\text{tam} = x[3] = 5$;



4.3. Sắp xếp chèn – Insertion sort

- Lượt chia 4: Chia ở vị trí $x[4]$, $\text{tam} = x[4] = -4$;



4.3. Sắp xếp chèn – Insertion sort

- Ví dụ 2

- Cho dãy số nguyên như sau:

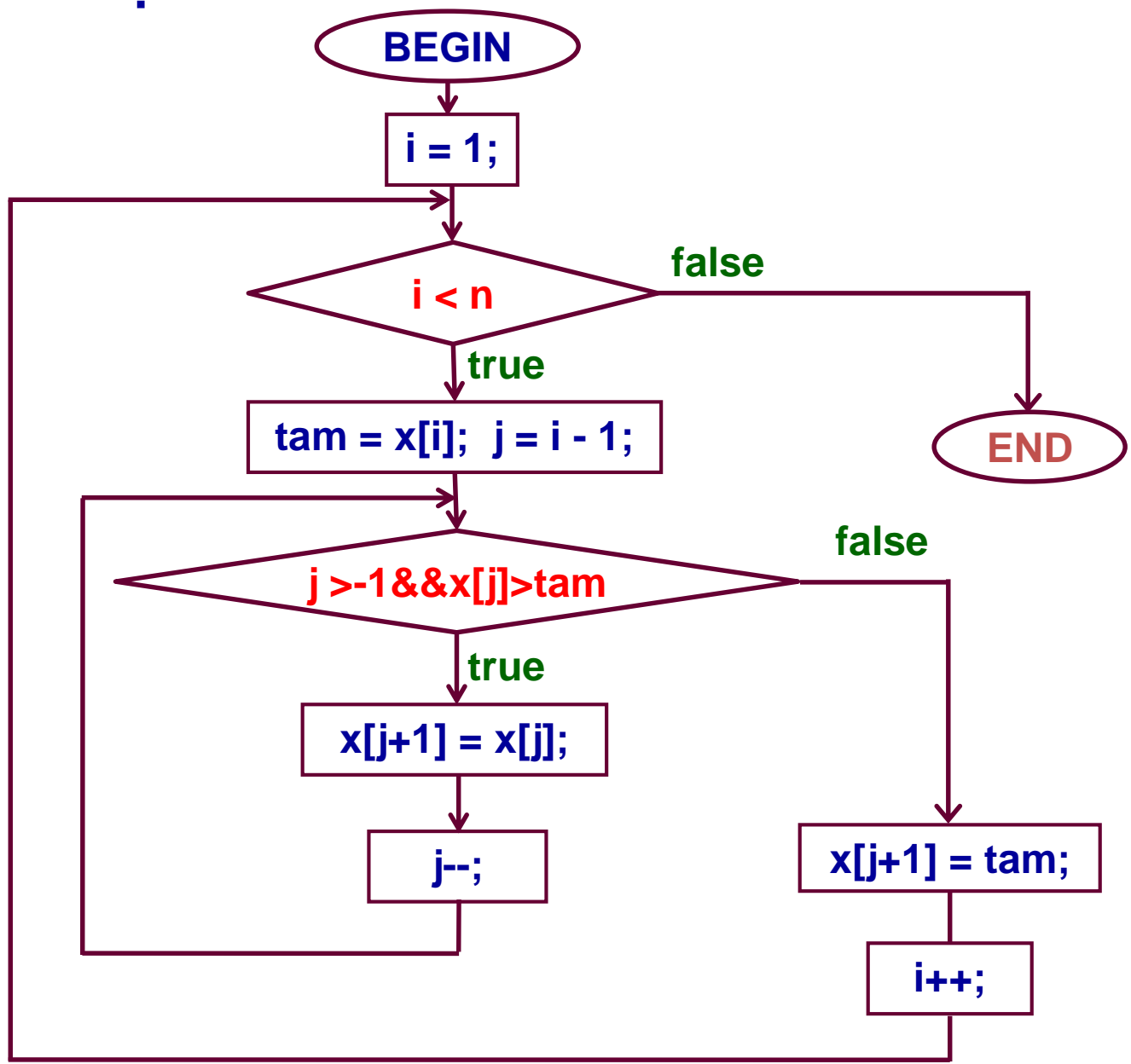
x[0]	x[1]	x[2]	x[3]	x[4]	x[5]
53	-21	67	15	60	40

- Yêu cầu: Dựa vào ý tưởng trên, minh họa việc sắp xếp dãy số theo chiều giảm dần.

4.3. Sắp xếp chèn – Insertion sort

- **Thiết kế giải thuật**
 - Thực hiện $n-1$ lần “chia” dãy thành dãy đích và dãy nguồn:
 - Dãy đích gồm các phần tử $X[0]$ đến $X[i-1]$ ($i=1 \rightarrow n-1$)
 - Dãy nguồn gồm các phần tử $X[i]$ đến $X[n-1]$
 - Mỗi lần chia lấy phần tử đầu của dãy nguồn (là $X[i]$ với $i=1 \rightarrow n-1$), chèn vào vị trí thích hợp trong dãy đích.

■ Lưu đồ giải thuật



4.3. Sắp xếp chèn – Insertion sort

- Giải thuật

```
void insertionSort(int x[], int n)
{
    for (int i=1; i<n; i++)
    {
        int tam = x[i];
        int j = i-1;
        while (j>-1 && x[j]>tam)
        {
            x[j+1] = x[j];
            j--;
        }
        x[j+1] = tam;
    }
}
```

Ví dụ cách biểu diễn kết quả sắp sếp chèn

Insertion Sort Execution Example



4.3. Sắp xếp chèn – Insertion sort

- Cài đặt ứng dụng thực hiện các yêu cầu:
 - Khởi tạo một số nguyên dương n .
 - Khởi tạo dãy có n số nguyên.
 - Sắp xếp dãy theo chiều tăng dần bằng giải thuật sắp xếp chèn.
 - In dãy vừa sắp ra màn hình.

4.3. Sắp xếp chèn – Insertion sort

- **Bài tập 1:** Cho dãy số nguyên

34 14 24 54 84 64 94 74 04

- Minh họa việc sắp xếp dãy số theo chiều tăng dần (giảm dần) bằng phương pháp sắp xếp chèn.
- Cài đặt chương trình sắp xếp dãy số.

- **Bài tập 2:** Cho dãy từ

John Wenger Anna Henry Thor Terry Ozil Adam Dennis

- Minh họa việc sắp xếp dãy từ theo trật tự từ điển (ngược lại với trật tự từ điển) bằng phương pháp sắp xếp chèn.
- Cài đặt chương trình sắp xếp dãy từ.

4.3. Sắp xếp chèn – Insertion sort

- **Bài tập 3: Cài đặt chương trình thực hiện các yêu cầu:**
 - Nhập vào một danh sách học sinh ($0 < n < 100$, n nhập từ bàn phím), mỗi học sinh gồm các thông tin: Mã học sinh, họ và tên, năm sinh và điểm trung bình.
 - Sắp xếp danh sách theo chiều tăng dần của tên học sinh bằng giải thuật sắp xếp chèn.
 - In danh sách vừa sắp ra màn hình.
 - Sắp xếp danh sách theo chiều giảm dần của điểm trung bình bằng thuật sắp xếp chèn.
 - In danh sách ra màn hình.