

# BÀI TOÁN SẮP XẾP

**Cho danh sách có  $n$  phần tử  $a_0, a_1, a_2, \dots, a_{n-1}$ .**

- Sắp xếp là quá trình xử lý các phần tử trong danh sách để đặt chúng theo một thứ tự thỏa mãn một số tiêu chuẩn nào đó dựa trên thông tin lưu tại mỗi phần tử như:
  - Sắp xếp danh sách lớp học tăng theo điểm TB.
  - Sắp xếp danh sách sinh viên tăng theo tên.
  - .....
- Để đơn giản trong việc trình bày giải thuật ta dùng mảng 1 chiều  $a$  để lưu danh sách trên trong bộ nhớ chính.

# CÁC THUẬT TOÁN SẮP XẾP ( 6 Tiết )

1. Phương pháp chọn
2. Phương pháp chèn
3. Nổi bọt
4. Đổi chỗ trực tiếp
5. Phương pháp trộn
6. Phương pháp đếm
7. Shell sort
8. Quick sort
9. Heap sort
10. Radix sort

# 1. THUẬT TOÁN INTERCHANGE SORT

## 1. BÀI TOÁN DẪN NHẬP

- Bài toán: Viết hàm liệt kê tất cả các cặp giá trị trong mảng một chiều các số nguyên. Lưu ý: các cặp trong mảng chỉ gặp nhau 1 lần

# 1. THUẬT TOÁN INTERCHANGE SORT

➤ Ví dụ: 

12	43	1	34	22
----	----	---	----	----

➤ Các cặp giá trị trong mảng là:

↪ (12,43), (12,01), (12,34), (12,22)

↪ (43,01), (43,34), (43,22)

↪ (01,34), (01,22)

↪ (34,22)

# 1. THUẬT TOÁN INTERCHANGE SORT

```
void LietKe (int a [ ], int n)
{
    for(int i=0; i<n-1; i++)
    {
        for(int j=i+1; j<n; j++)
        {
            printf("(%d,%d)", a[i],a[j]);
        }
    }
}
```

# 1. THUẬT TOÁN INTERCHANGE SORT

## 4. ÁP DỤNG THUẬT TOÁN

- Cho dãy số a:

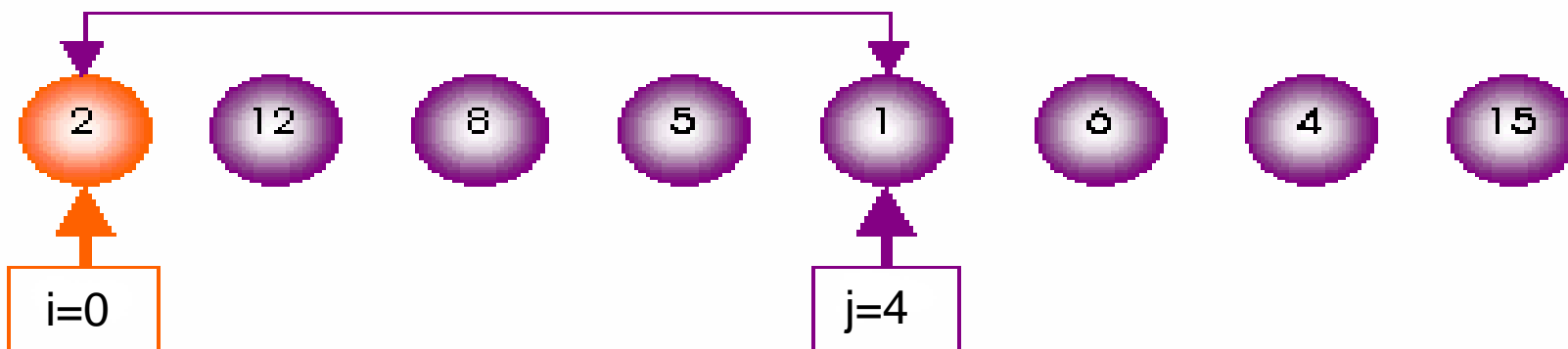
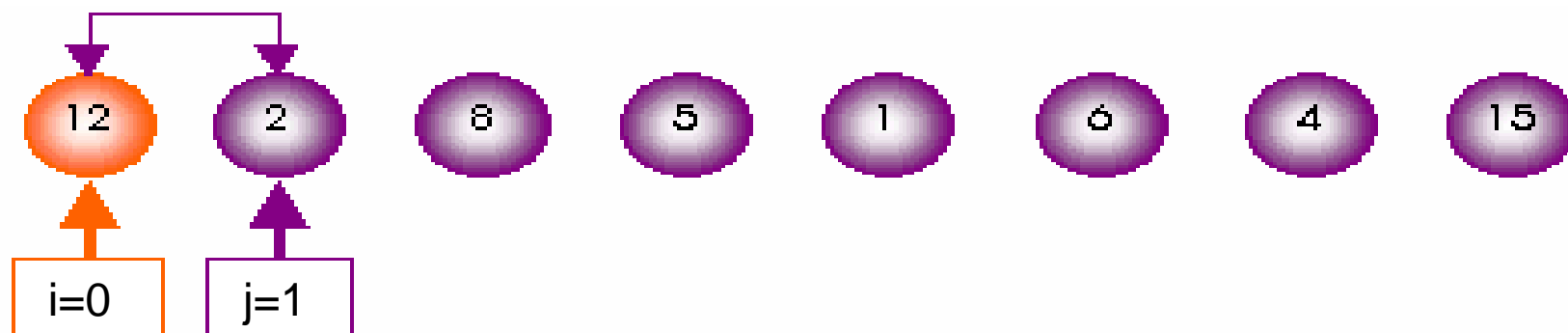
12      2      8      5      1      6      4      15

- Hãy sắp xếp dãy trên tăng dần
- Thứ tự các bước khi sắp tăng dần mảng trên bằng thuật toán interchange sort như sau :

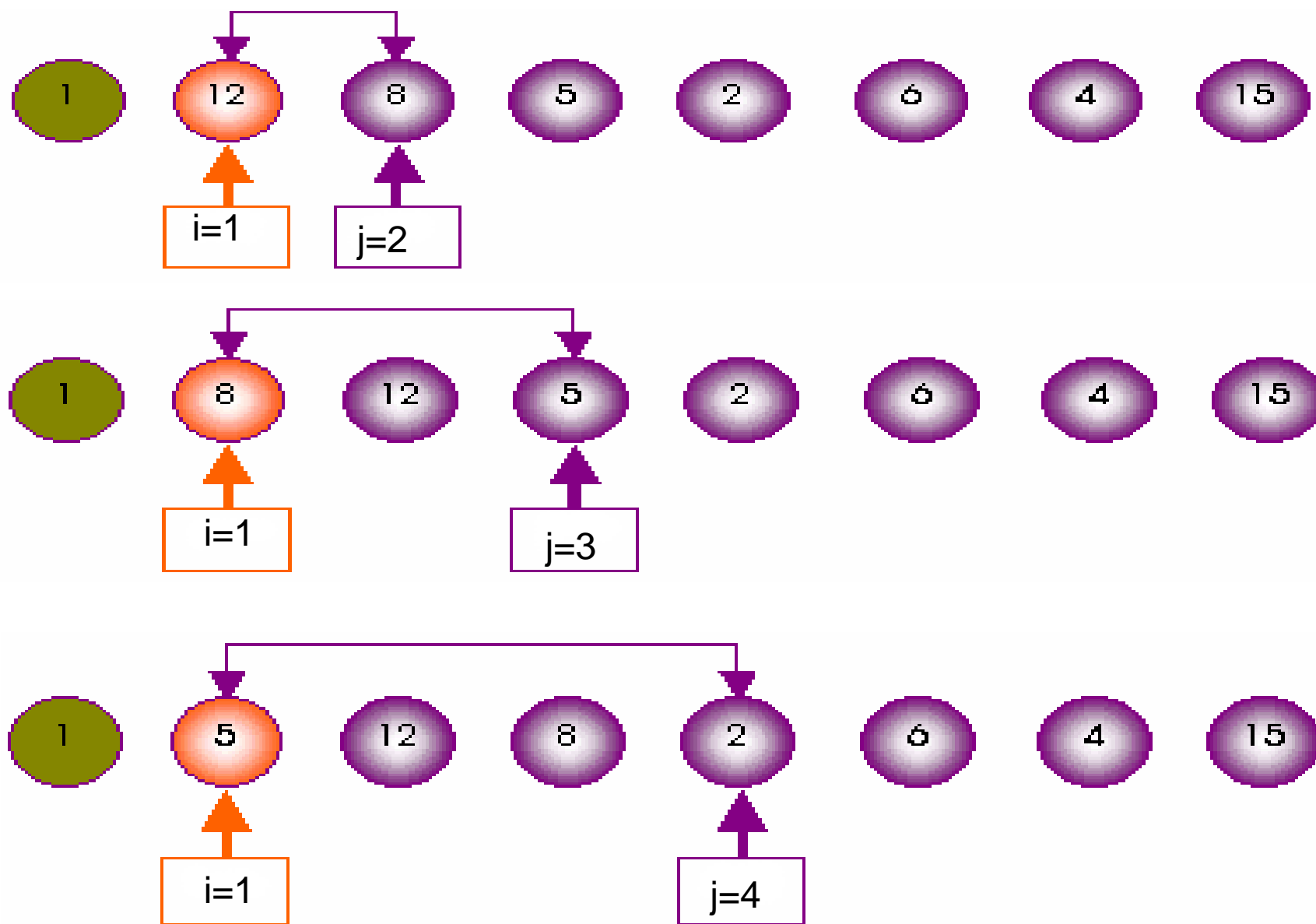
# Đổi Chỗ Trục Tiếp – Interchange Sort

➤ Cho dãy số a:

12    2    8    5    1    6    4    15

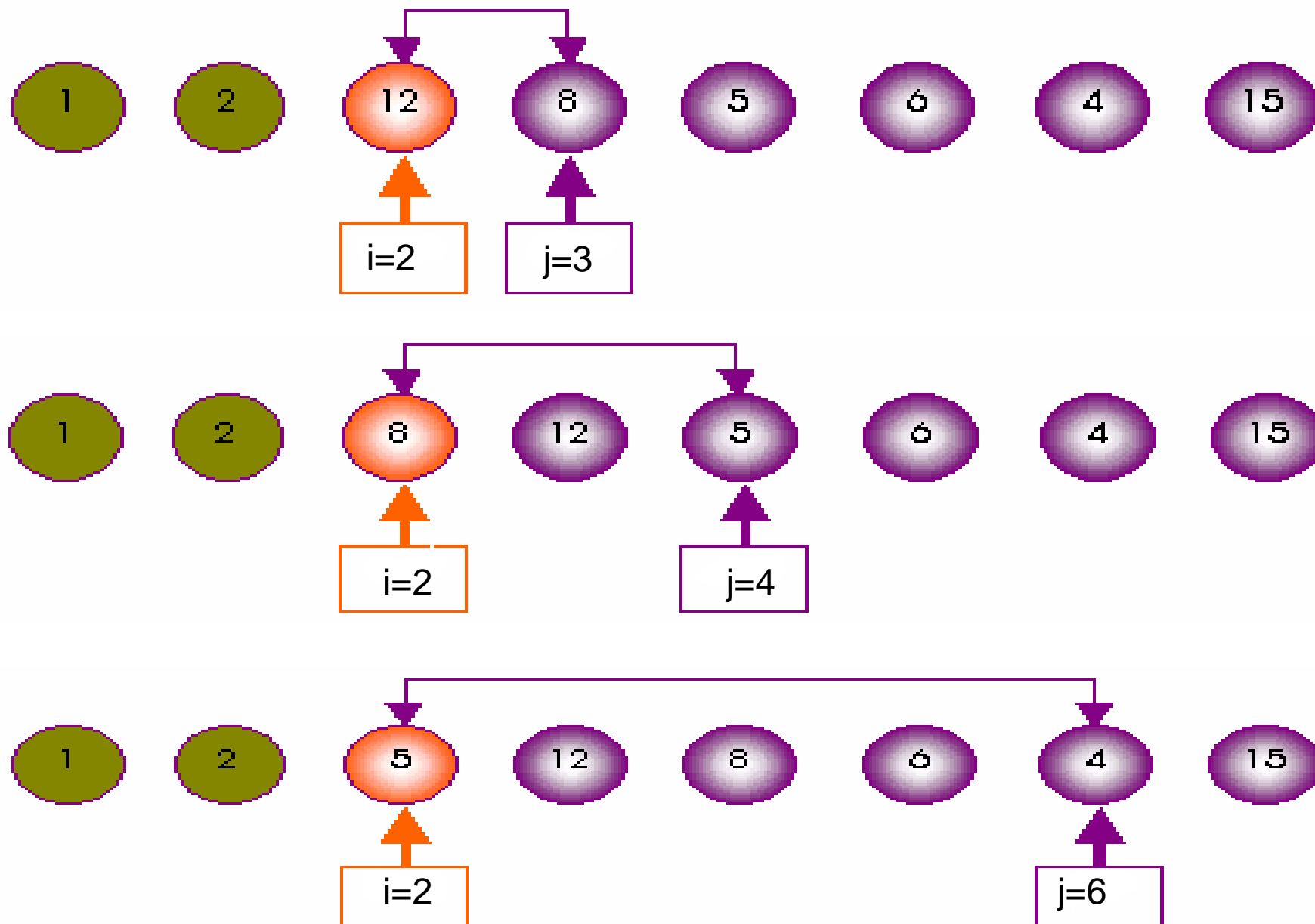


# Đổi Chỗ Trục Tiếp – Interchange Sort

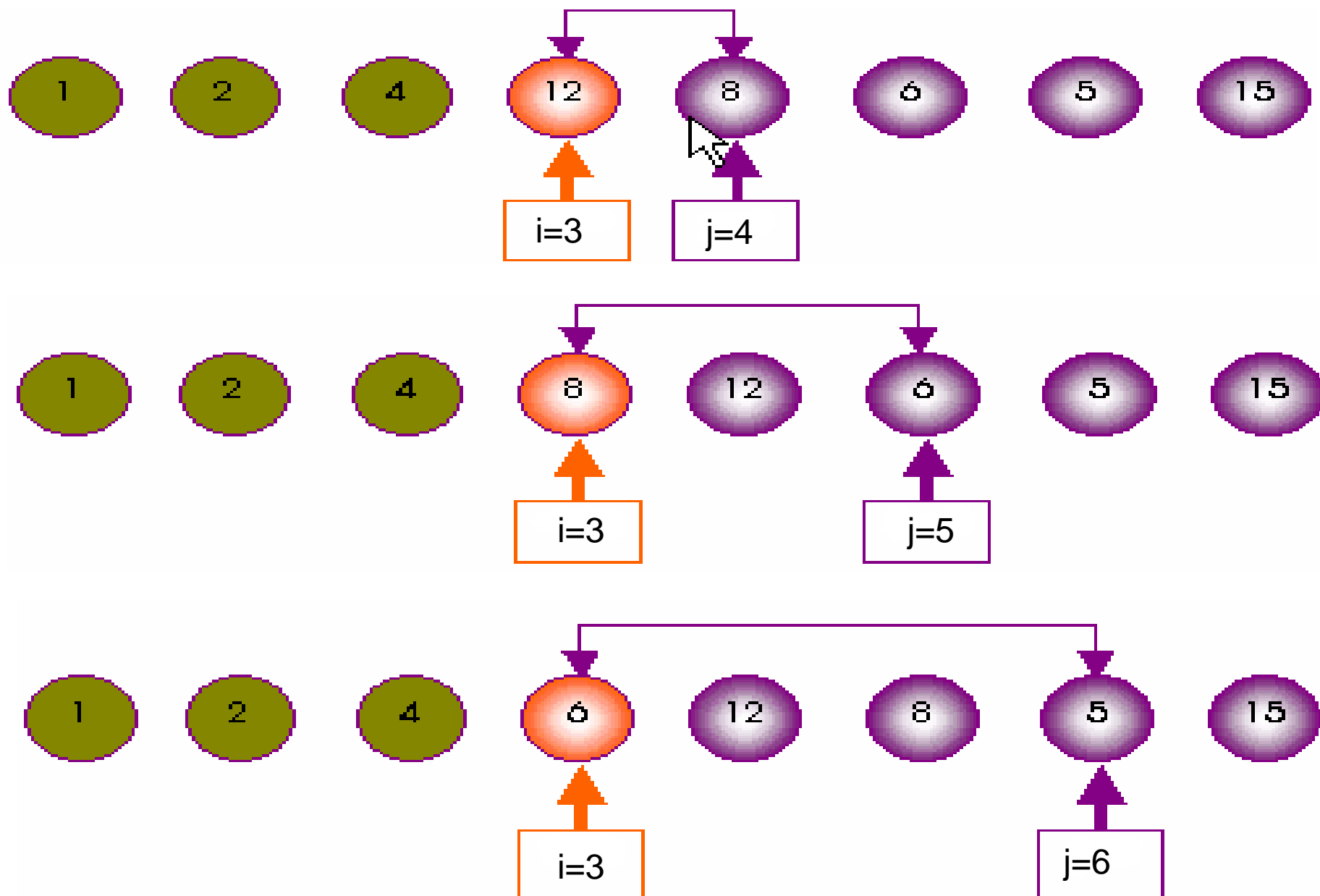




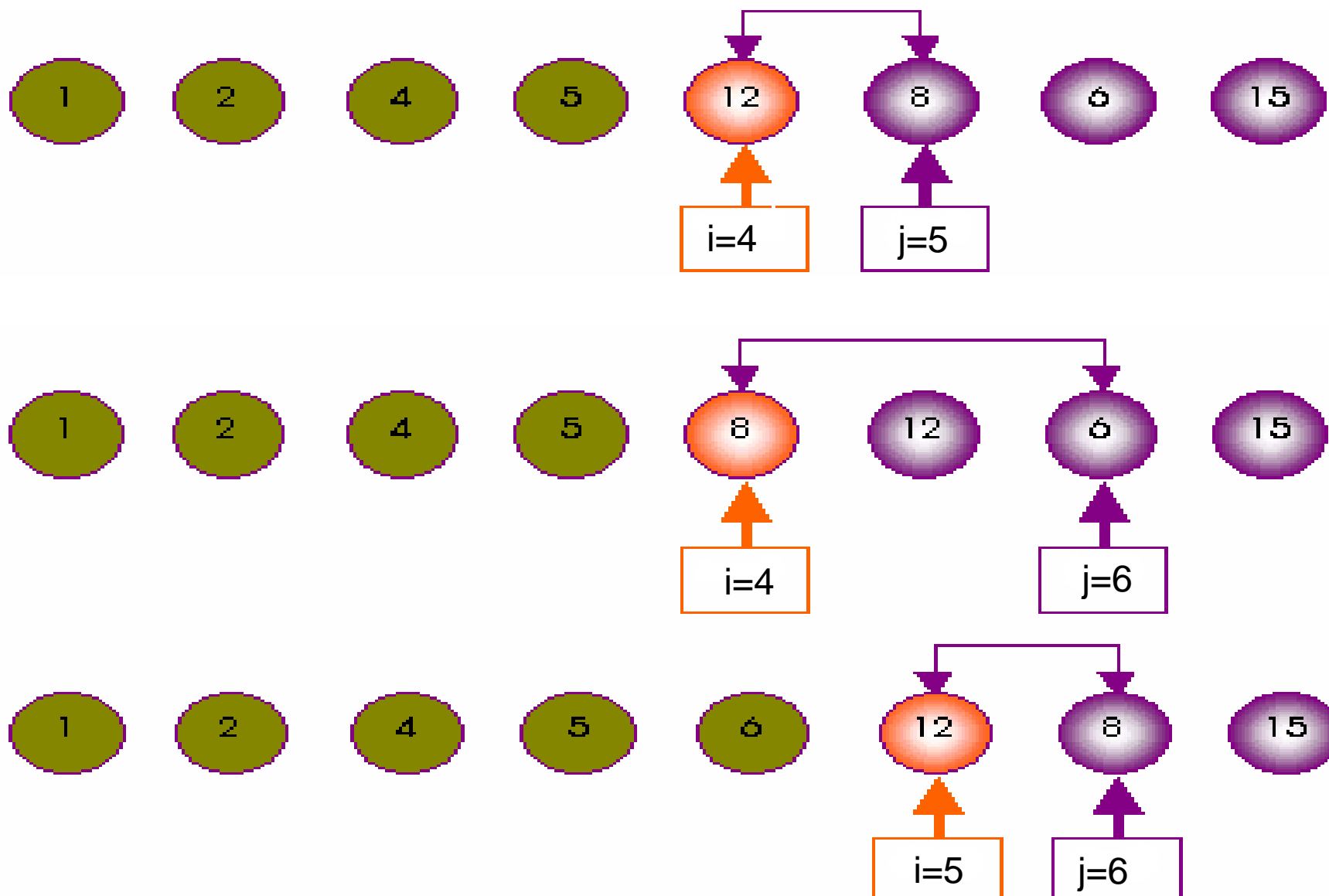
# Đổi Chỗ Trục Tiếp – Interchange Sort



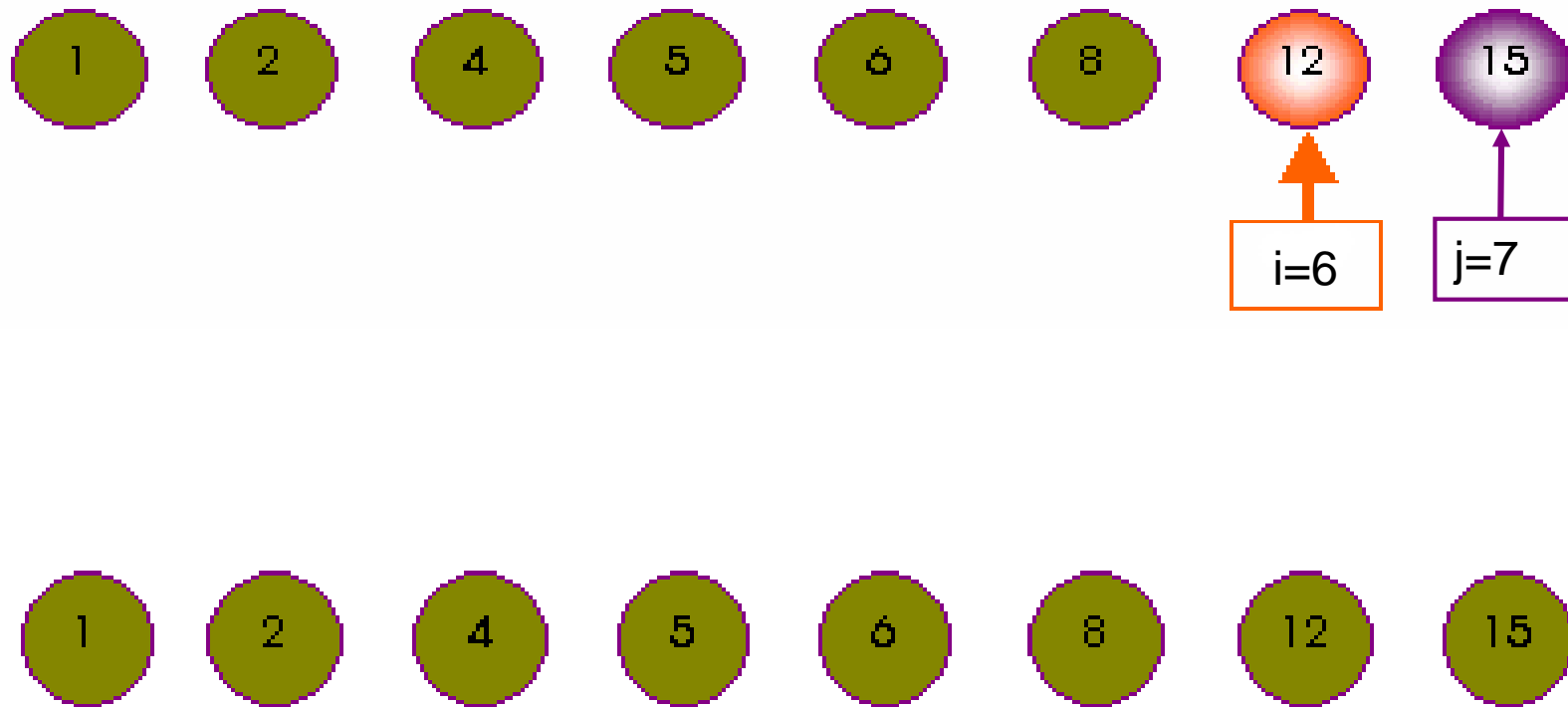
# Đổi Chỗ Trục Tiếp – Interchange Sort



# Đổi Chỗ Trực Tiếp – Interchange Sort



# Đổi Chỗ Trục Tiếp – Interchange Sort



# 1. THUẬT TOÁN INTERCHANGE SORT

## 3. TƯ TƯỞNG THUẬT TOÁN

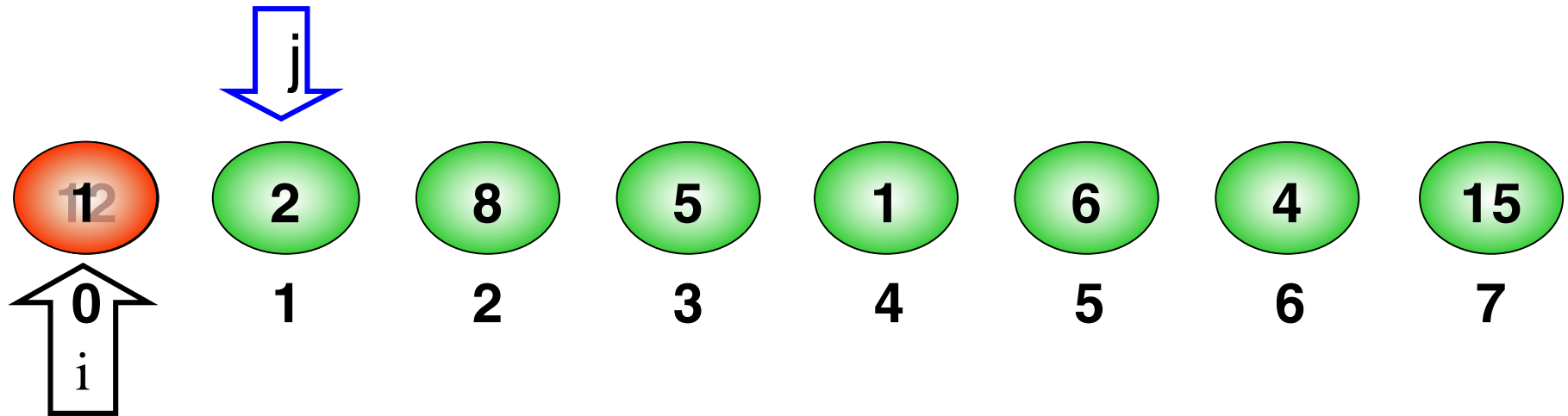
- Thuật toán interchange sort sẽ duyệt qua tất cả các cặp giá trị trong mảng và hoán vị hai giá trị trong một cặp nếu cặp giá trị đó là nghịch thế.

# Các bước Tiến hành

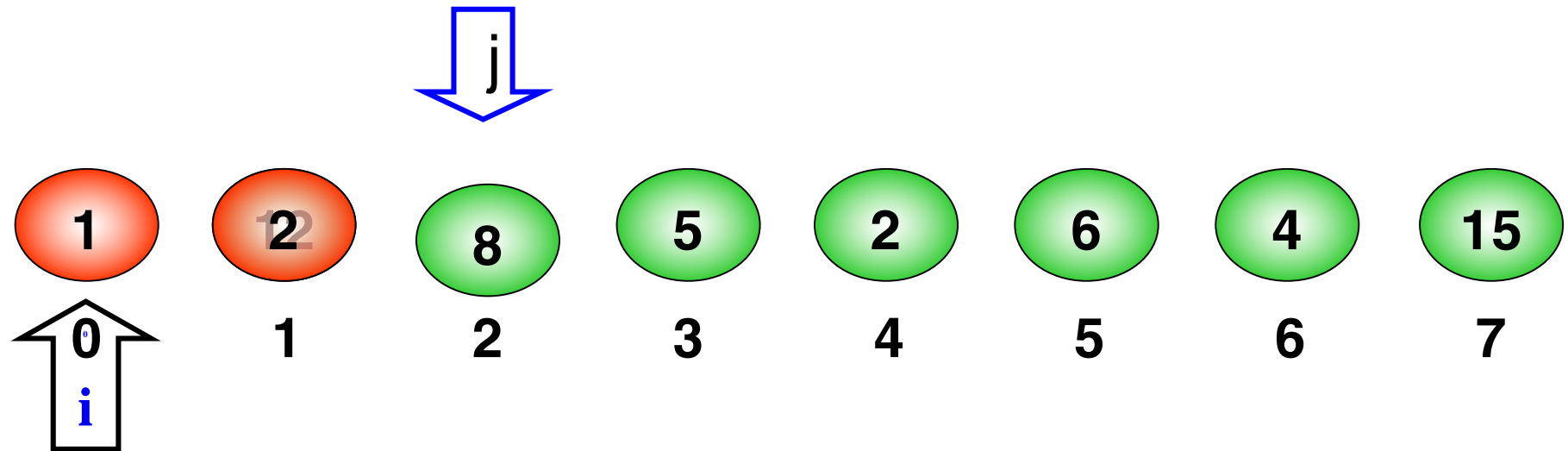
- Bước 1:  $i = 0$ ;      *// bắt đầu từ đầu dãy*
- Bước 2:  $j = i+1$ ; *// tìm các nghịch thế với  $a[i]$*
- Bước 3:  
    Trong khi  $j < n$  thực hiện  
        Nếu  $a[j] < a[i]$  *// xét cặp  $a[i], a[j]$*   
            hoanvi( $a[i], a[j]$ );  
         $j = j+1$ ;
- Bước 4:  $i = i+1$ ;  
    Nếu  $i < n-1$ : Lặp lại Bước 2.  
    Ngược lại: Dừng.

# 1. THUẬT TOÁN INTERCHANGE SORT

```
void InterchangeSort( int a[ ], int n )
{
    for ( int i = 0 ; i < n-1 ; i++ )
        for ( int j = i+1 ; j < n ; j++ )
            if( a[ i ] > a[ j ] )
            {
                int tam = a[i] ;
                a[i] = a[j] ;
                a[j] = tam;
            }
}
```







# Độ Phức Tạp Của Thuật Toán

Trường hợp	Số lần so sánh	Số lần hoán vị
Tốt nhất	$\sum_{i=1}^{n-1} (n-i+1) = \frac{n(n-1)}{2}$	0
Xấu nhất	$\frac{n(n-1)}{2}$	$\sum_{i=1}^{n-1} (n-i+1) = \frac{n(n-1)}{2}$

# CÁC THUẬT TOÁN SẮP XẾP ( 6 Tiết )

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