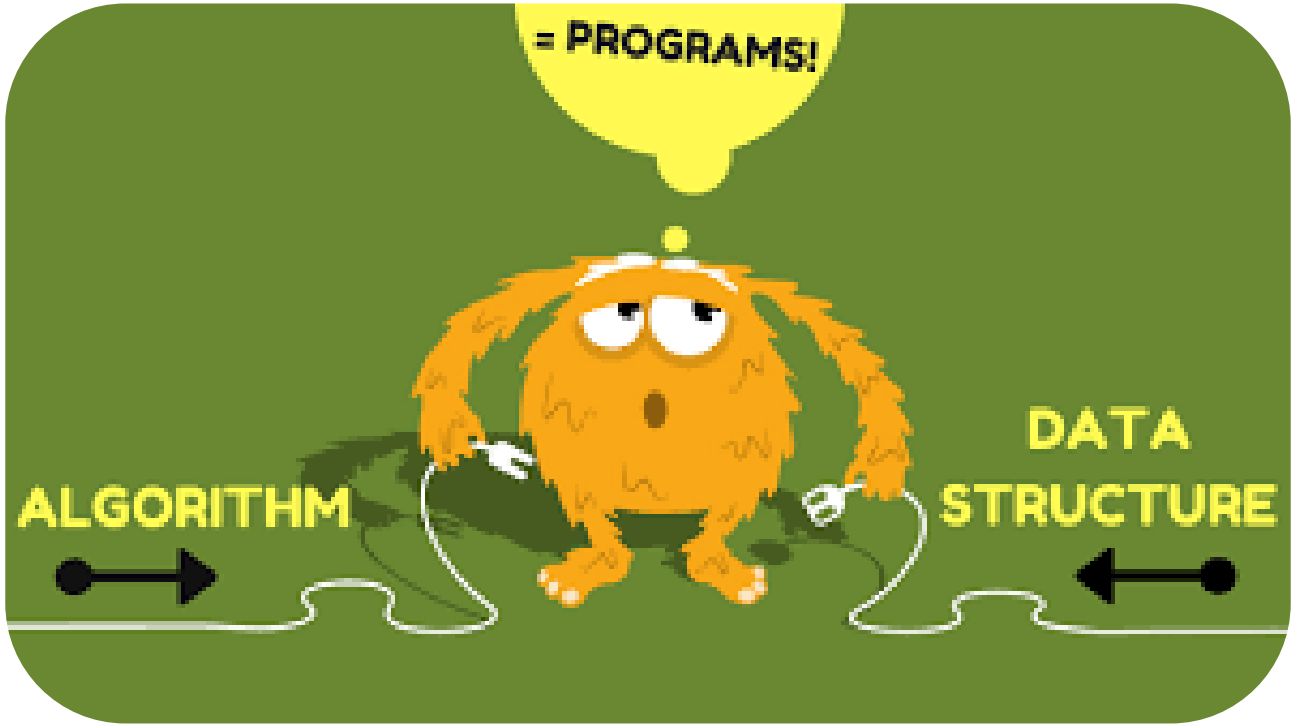
CÁC THUẬT TOÁN SẮP XẾP



MỤC TIÊU BÀI HỌC

- Tìm hiểu các giải thuật sắp xếp cơ bản trên cấu trúc dữ liệu mảng.
- Đánh giá và so sánh hiệu quả các giải thuật.

NỘI DUNG BÀI HỌC

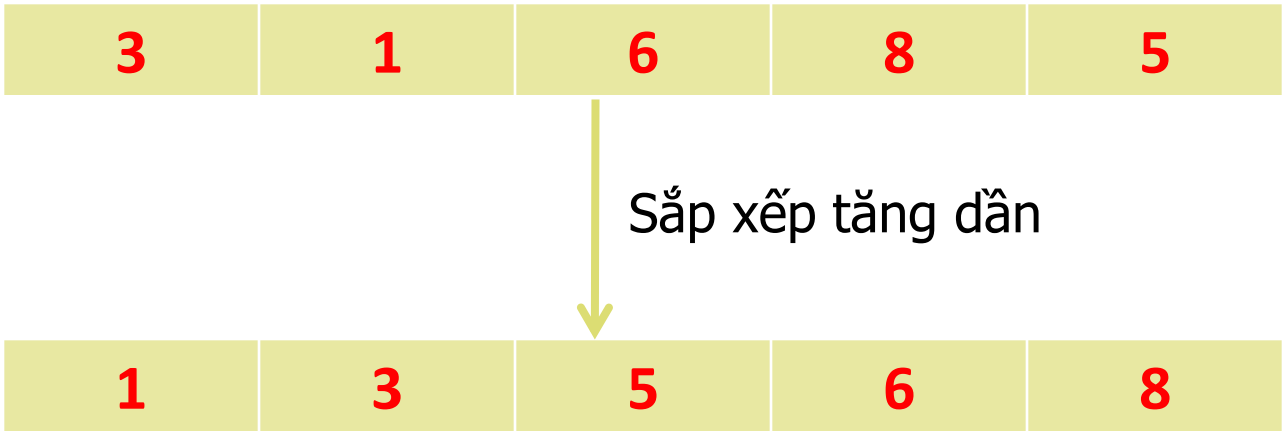
1. Bài toán sắp xếp
2. Sắp xếp chọn
3. Sắp xếp nổi bọt
4. Sắp xếp chèn

1

BÀI TOÁN SẮP XẾP

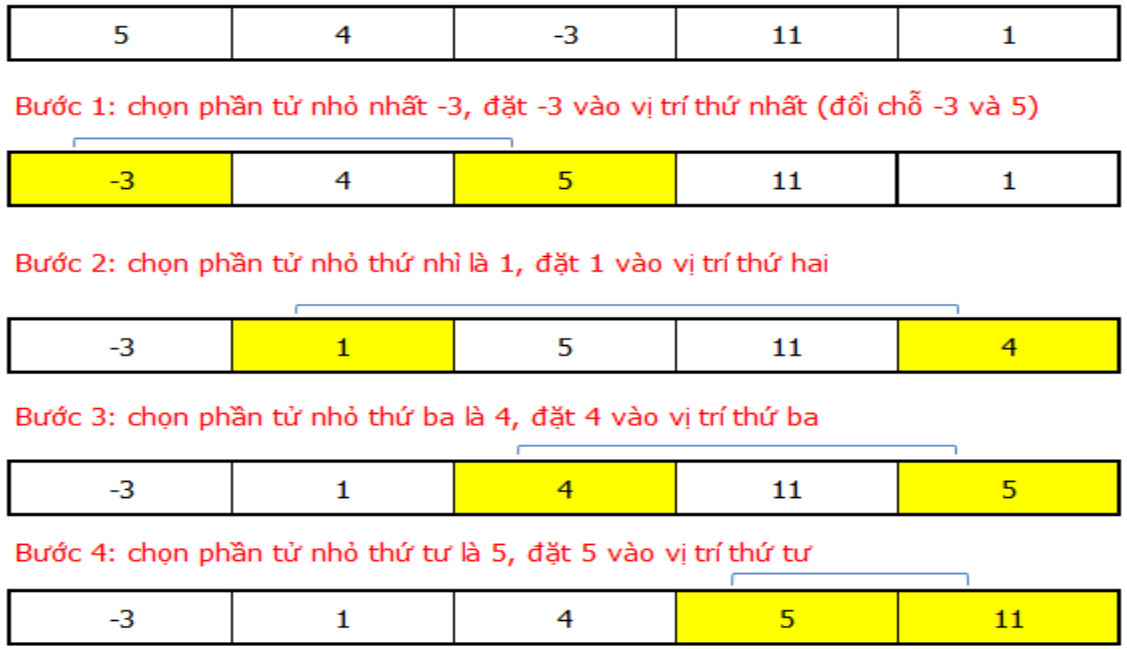
BÀI TOÁN SẮP XẾP DÃY SỐ

- Sắp xếp là quá trình xử lý một danh sách các phần tử để đặt chúng theo một thứ tự nào đó (tăng dần, giảm dần) dựa trên nội dung thông tin lưu giữ tại mỗi phần tử.



BÀI TOÁN SẮP XẾP DÃY SỐ

- Bài toán: Cho trước một dãy số $a_1, a_2, \ldots, a_N$ được lưu trữ trong cấu trúc dữ liệu mảng.
- Sắp xếp dãy số $a_1, a_2, \ldots, a_N$ là thực hiện việc bố trí lại các phần tử sao cho hình thành được dãy mới $a_{k1}, a_{k2}, \ldots, a_{kN}$ có thứ tự (ví dụ thứ tự tăng) nghĩa là $a_{ki} > a_{ki-1}$.



BÀI TOÁN SẮP XẾP DÃY SỐ

- Để quyết định được những 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 của hầu hết các thuật toán sắp xếp.
- Chú ý: Khi xây dựng một thuật toán sắp xếp cần tìm cách giảm thiểu những phép so sánh và đổi chỗ không cần thiết để tăng hiệu quả của thuật toán

2 THUẬT TOÁN SẮP XẾP LỰA CHỌN

SẮP XẾP LỰA CHỌN

- Ý tưởng:
  - ✓ Chọn phần tử nhỏ nhất trong N phần tử ban đầu, đưa phần tử này về vị trí đầu dãy hiện hành; sau đó loại nó khỏi danh sách sắp xếp tiếp theo.
  - ✓ Xem dãy hiện hành chỉ còn N-1 phần tử của dãy ban đầu, bắt đầu từ vị trí thứ 2; lặp lại quá trình trên cho dãy hiện hành… đến khi dãy hiện hành chỉ còn 1 phần tử.

SẮP XẾP LỰA CHỌN

- Các bước sắp xếp tăng dần:

Bước 1: i = 1 // lần xử lý đầu tiên

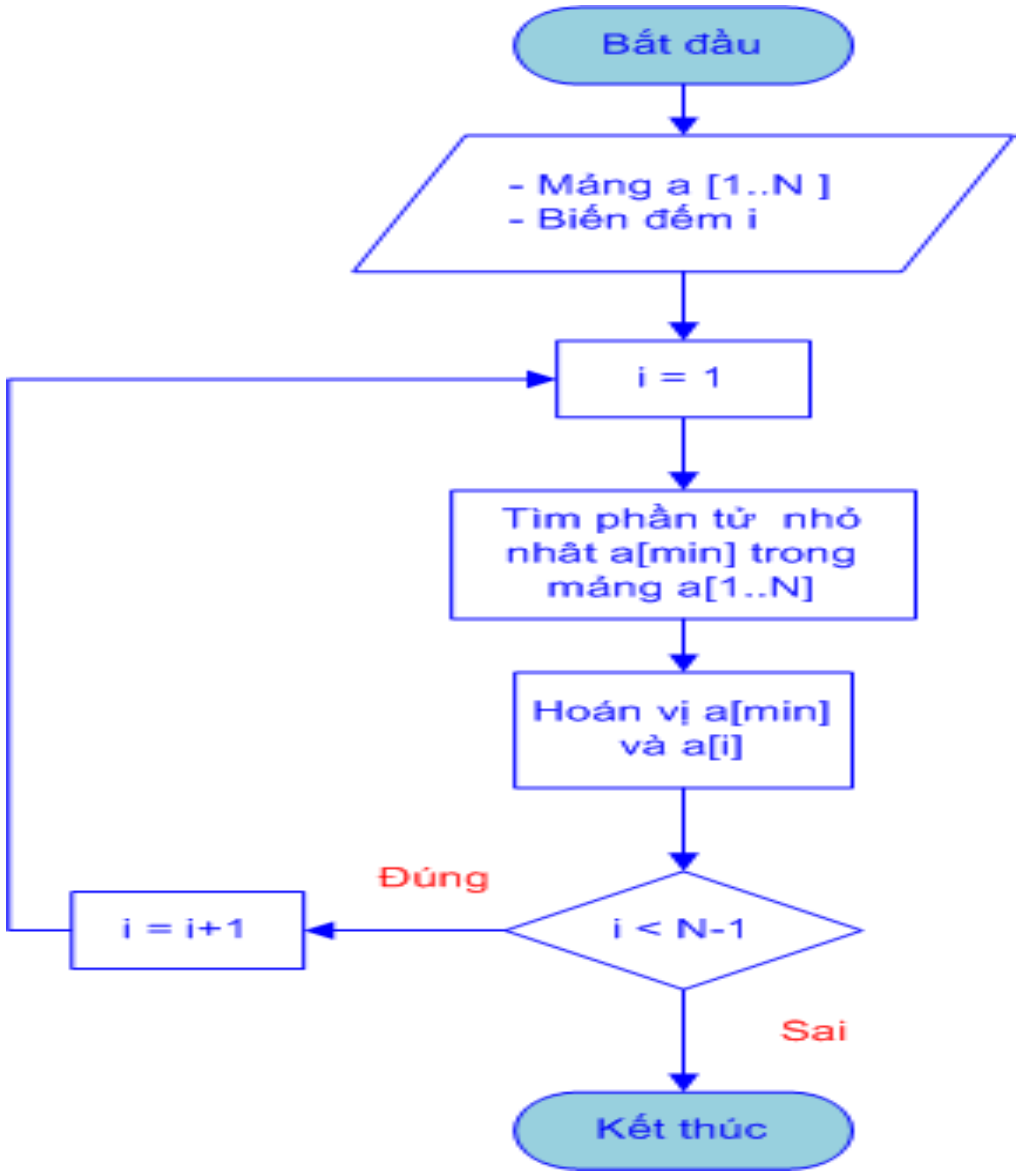
Bước 2: Tìm phần tử nhỏ nhất a[min] trong dãy hiện hành từ a[i] đến a[N]

Bước 3: Hoán vị a[min] và a[i]

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

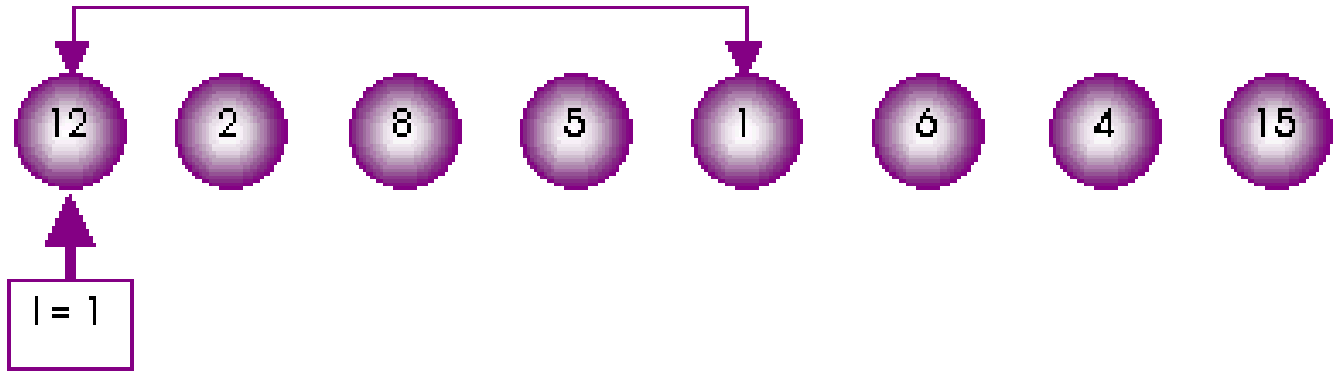
Ngược lại: Dừng

- Lưu đồ thuật toán:



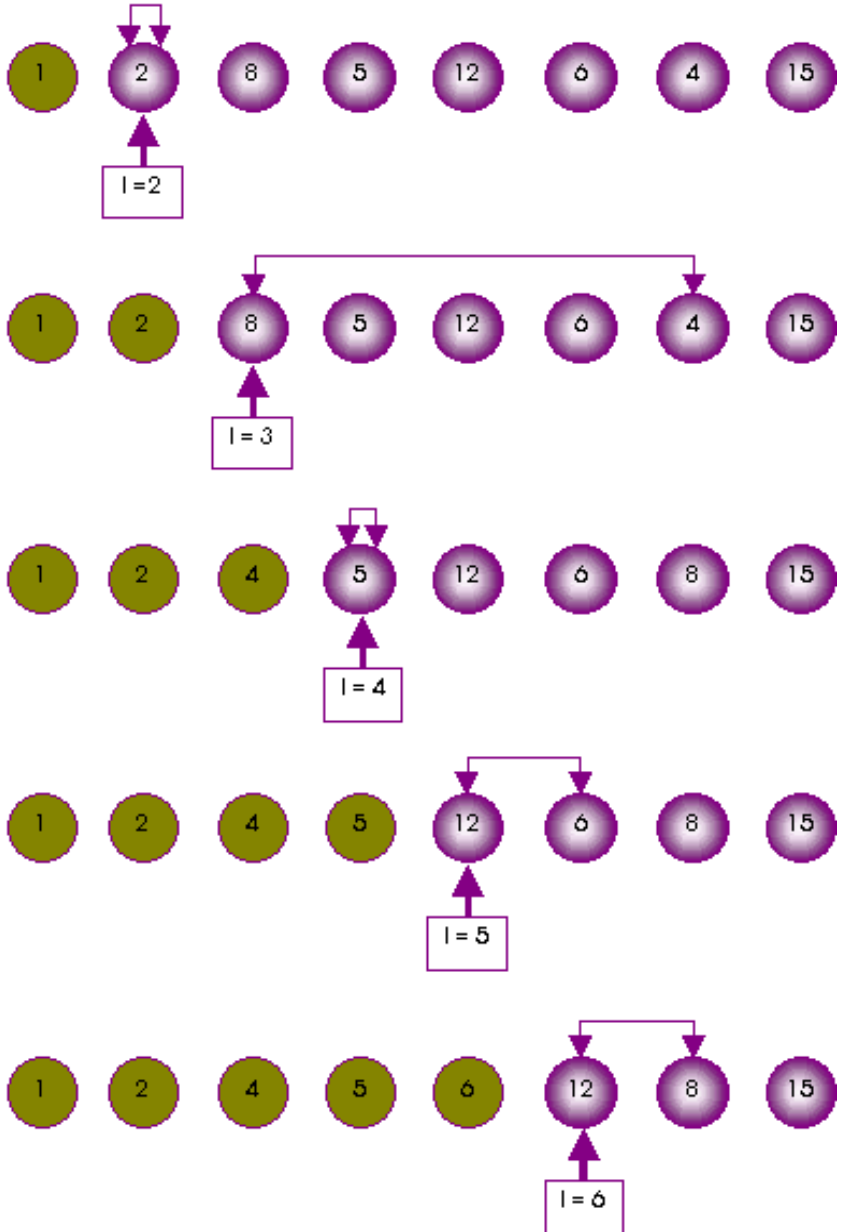
SẮP XẾP LỰA CHỌN

- Ví dụ minh họa:
  - Cho dãy số a: **{12, 2, 8, 5, 1, 6, 4, 15 }**



SẮP XẾP LỰA CHỌN

- Ví dụ minh họa(tt):



SẮP XẾP LỰA CHỌN

- Cài đặt thuật toán:

```
void selectionSort(int A[], int N) {
    int m;
    for (int i=0; i < N-1; i++){
        m=i;
        for (int k=i+1; k < N; k++)
            if (A[k] < A[m]) m=k;
        if (m != i) swap(A[i],A[m]);
    }
}
```

SẮP XẾP LỰA CHỌN

- Đánh giá giải thuật trên: Ở lượt thứ i, bao giờ cũng cần (n-i) lần so sánh để xác định phần tử nhỏ nhất hiện hành. Do vậy số lần so sánh:

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

- Số lần hoán vị (một hoán vị bằng 3 phép gán) lại phụ thuộc vào tình trạng ban đầu của dãy số, ta chỉ có thể ước lược trong từng trường hợp như sau:

Trường hợp	Số lần so sánh	Số phép gán
Tốt nhất	n(n-1)/2	3(n-1)
Xấu nhất	n(n-1)/2	n(n-1)/2 + 3(n-1)

3

THUẬT TOÁN SẮP XẾP NỔI BỌT

SẮP XẾP NỔI BỌT

- Ý tưởng: Xuất phát từ đầu dãy, so sánh 2 phần tử cạnh nhau để đưa phần tử nhỏ hơn lên trước, sau đó lại xét cặp tiếp theo cho đến khi tiến về đầu dãy. Nhờ vậy, ở lần xử lý thứ i sẽ tìm được phần tử ở vị trí đầu dãy là i.
- Các bước:

Bước 1: i=1 // lần xử lý đầu tiên

Bước 2: j=N // duyệt từ cuối dãy trở về vị trí

Trong khi j>i thực hiện:

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

j= j-1

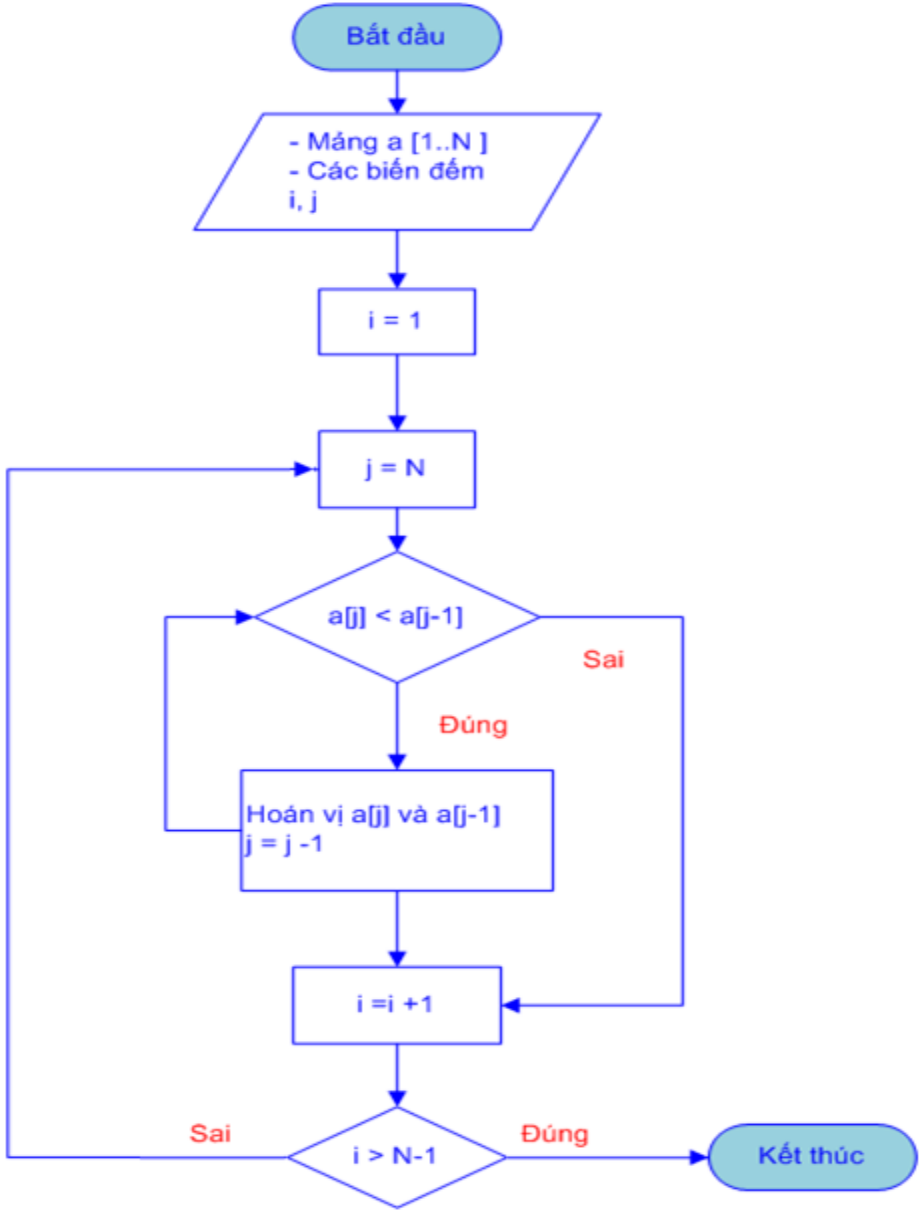
Bước 3: i=i+1 // lần xử lý tiếp theo

Nếu i>N-1 thì dừng

Ngược lại, lặp lại Bước 2

SẮP XẾP NỔI BỌT

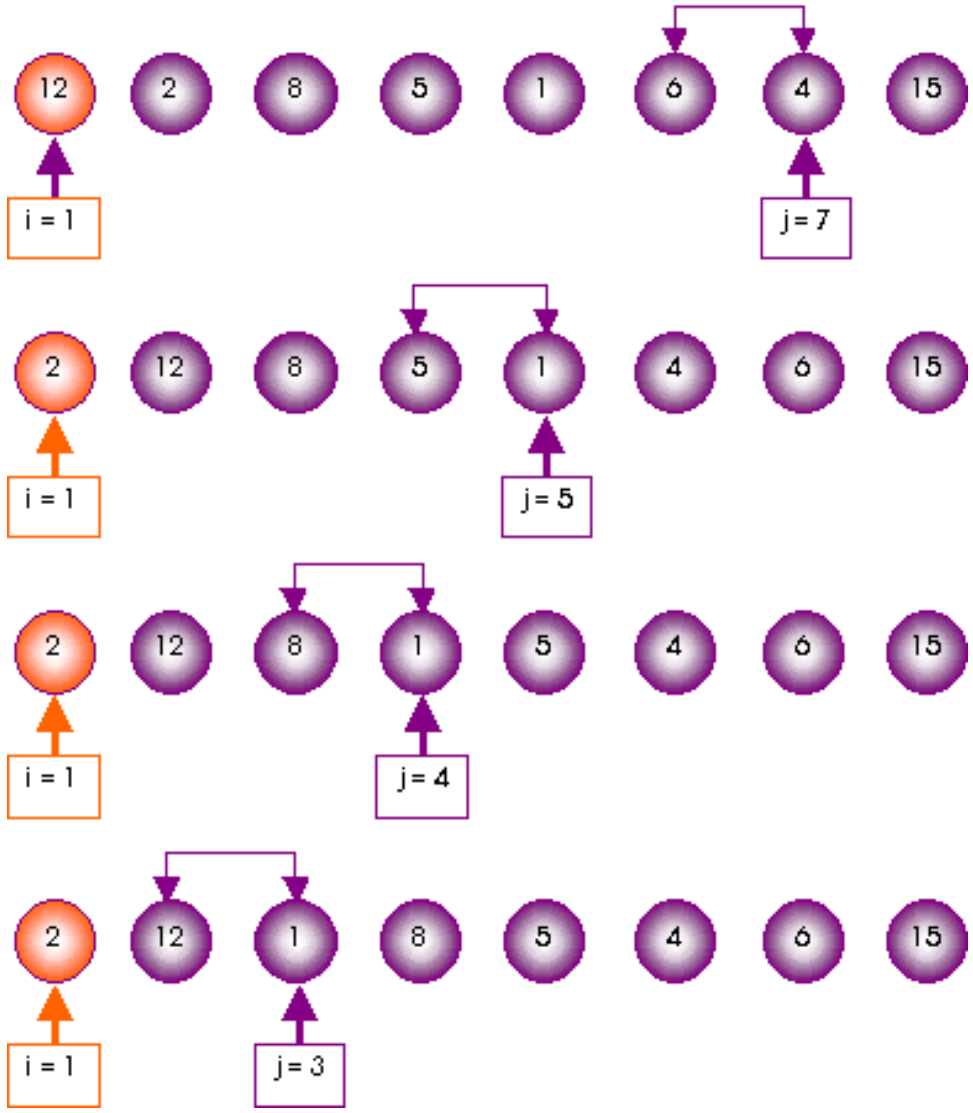
- Lưu đồ thuật toán:



SẮP XẾP NỔI BỌT

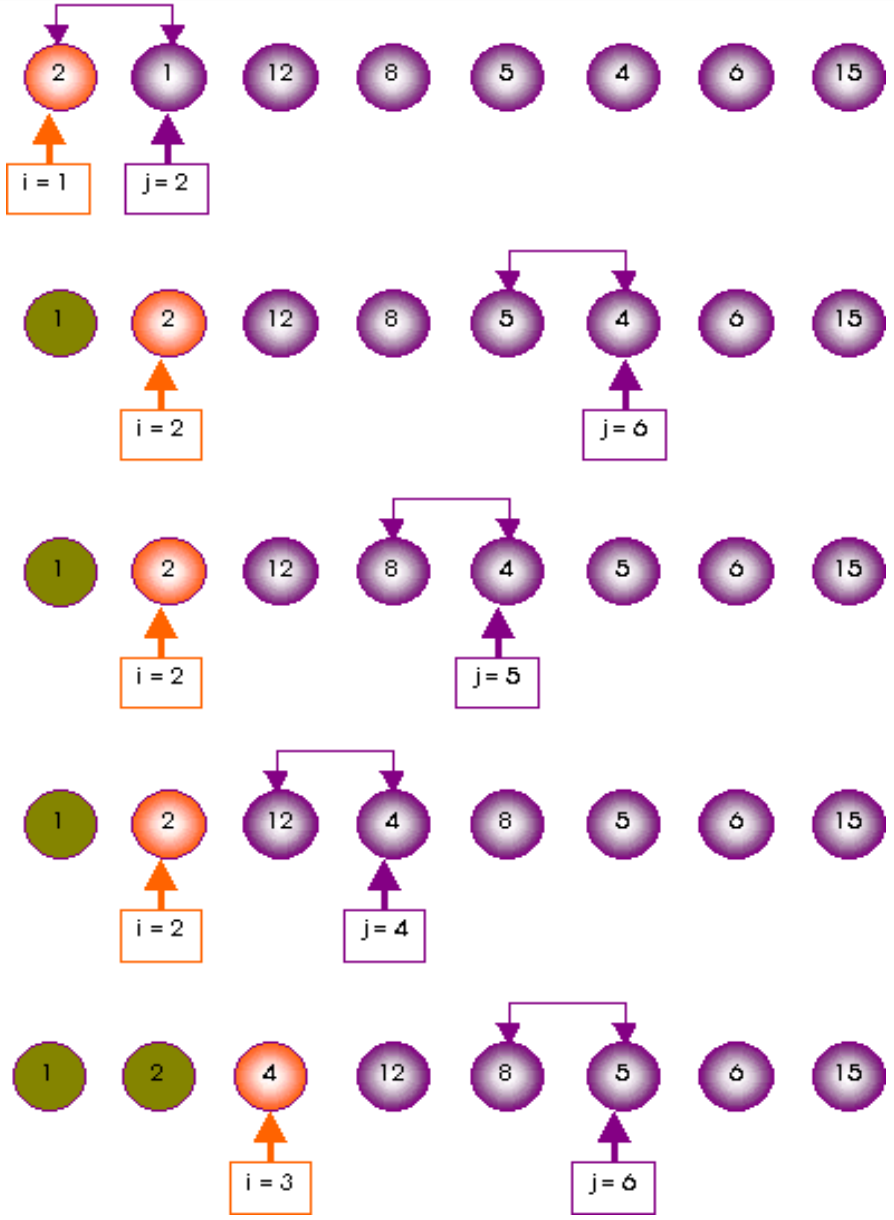
- Ví dụ:

  Cho dãy số a: **{12, 2, 8, 5, 1, 6, 4, 15 }**



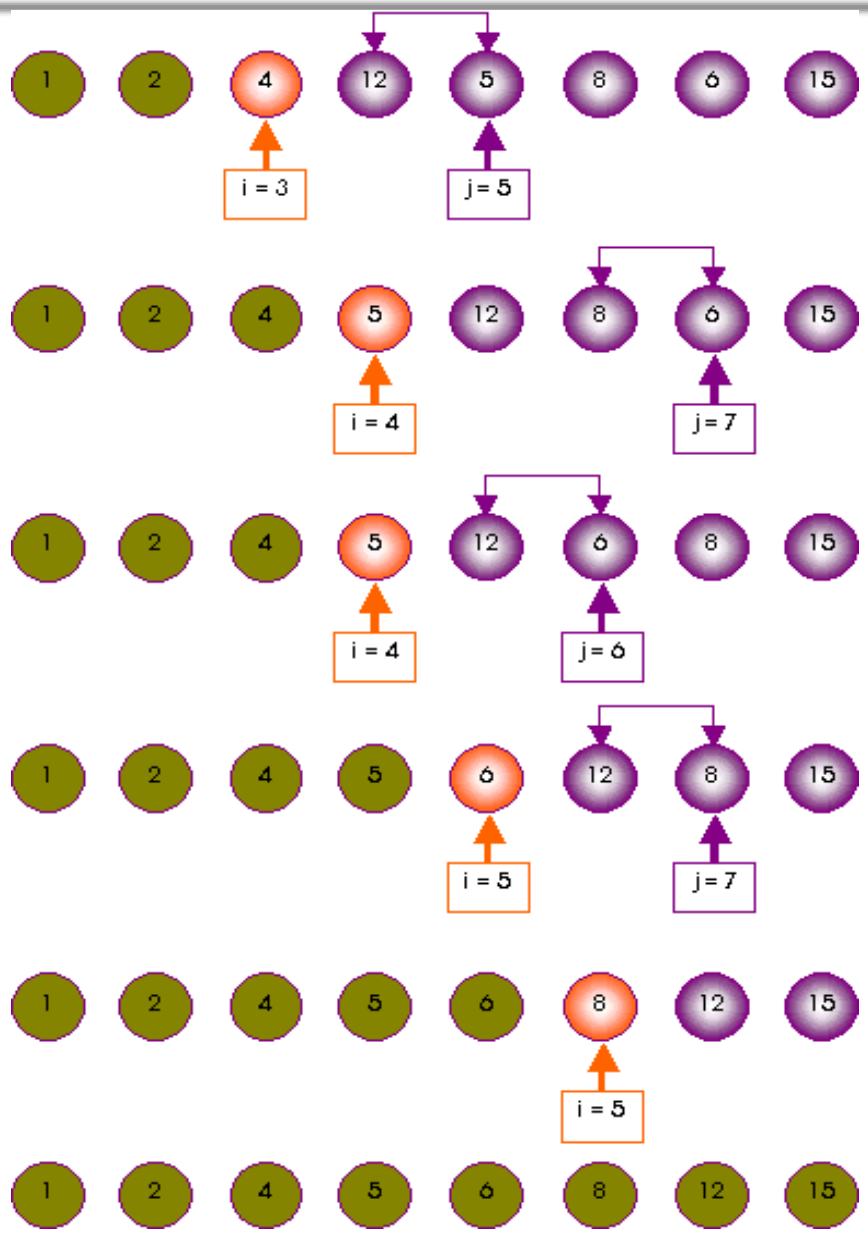
SẮP XẾP NỔI BỌT

- Ví dụ (tt):



SẮP XẾP NỔI BỌT

- Ví dụ (tt):



SẮP XẾP NỔI BỌT

- Cài đặt giải thuật

```
1 void swap(int &a, int &b)
2 {
3     int c = a;
4     a = b;
5     b = c;
6 }
```

```
1 void Sapxep(int a[], int n)
2 {
3     int i, j;
4     for (int i = 0; i<n - 2; i++)
5         for (int j = n - 1; j>i; j--)
6             if (a[j] < a[j - 1])
7                 swap(a[j], a[j - 1]);
8 }
```

Đánh giá giải thuật:

- Ở lượt thứ i, bao giờ cũng cần (n-i+1) lần so sánh để xác định phần tử nhỏ nhất hiện hành.

  Do vậy số lần so sánh:

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

- Số lượng phép hoán vị thực hiện tùy thuộc vào kết quả so sánh, có thể ước lược trong từng trường hợp như sau

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

4

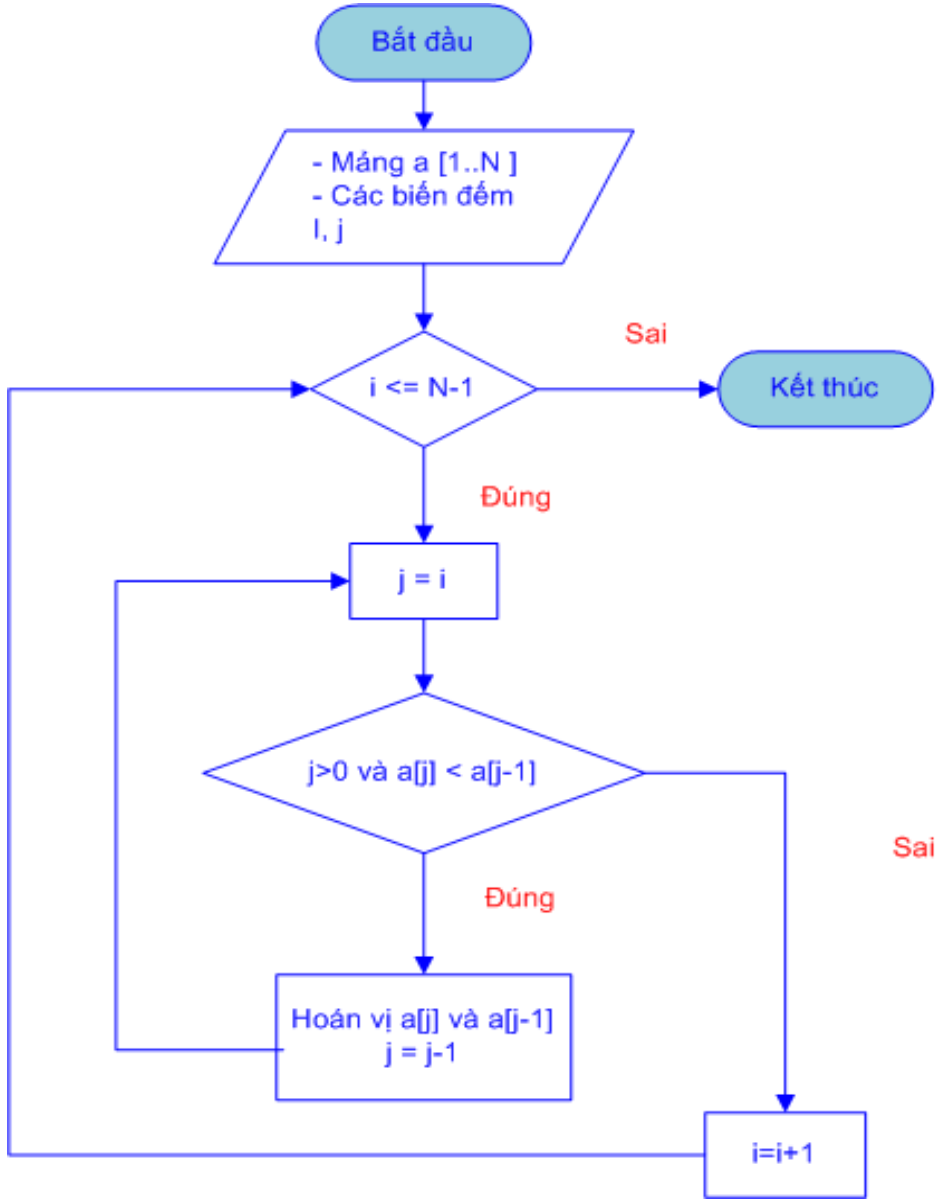
THUẬT TOÁN SẮP XẾP CHÈN

SẮP XẾP CHÈN

➢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] được sắp 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] được sắp thứ tự
- ✓ Tổng quát bước i, chèn phần tử a[i] vào đúng vị trí trong dãy đã sắp xếp a[0],... a[i-1] sao cho dãy a[0], a[1],... a[i] được sắp thứ tự.
- ✓ Sau N-1 bước thì kết thúc

- Lưu đồ thuật toán

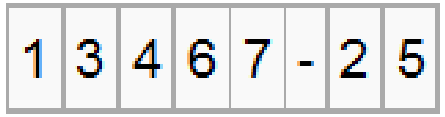


SẮP XẾP CHÈN

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

- Để tiếp tục sắp xếp phần tử thứ tư $a_4 = 6$ vào danh sách con đó, ta tìm vị trí thích hợp của nó là sau 3 và trước 7.

- Làm tiếp theo với $a_5 = 4$ ta được



- Làm tiếp theo với a6 = 2 ta được

- Cuối cùng chèn a7 = 5

- Cài đặt giải thuật

```
for (i=1;i<N;i++){
   k = i;
   b = a_{i+1};
   while (k>=1 AND b<a_k){
         a_{k+1} = a_k;
         k--;
   }
   a_{k+1} = b;
}
```

- Đánh giá giải thuật: Độ phức tạp giải thuật phụ thuộc vào số lần so sánh. Ở lượt thứ i, tối đa cần i lần so sánh để tìm được vị trí chèn thích hợp. Do vậy số lần so sánh tối đa là:

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

TỔNG KẾT

- Sắp xếp là những bài toán quan trọng trong lập trình máy tính.
- Việc sắp xếp thường được tiến hành trên mảng.


感謝你
Danke
Gracias
Grazie
Merci
Thanks!
THANK
YOU!


VAIO