**Insertion-sort**:

**Bước 1** - Nếu phần tử là phần tử đầu tiên, giả sử rằng nó đã được sắp xếp. Trả về 1.

**Bước 2** - Chọn phần tử tiếp theo và gán nó trong key.

**Bước 3** - Bây giờ, so sánh key với tất cả các phần tử trong mảng đã sắp xếp.

**Bước 4** - Nếu phần tử trong mảng đã sắp xếp nhỏ hơn phần tử hiện tại, chuyển sang phần tử tiếp theo. Ngược lại, chuyển các phần tử lớn hơn trong mảng sang bên phải.

**Bước 5** - Chèn giá trị key.

**Bước 6** - Lặp lại cho đến khi mảng được sắp xếp.

Ví dụ step by step:



**Bubble sort**:

Thuật toán sắp xếp bubble sort thực hiện sắp xếp dãy số bằng cách lặp lại công việc đổi chỗ 2 số liên tiếp nhau nếu chúng đứng sai thứ tự(số sau bé hơn số trước với trường hợp sắp xếp tăng dần) cho đến khi dãy số được sắp xếp

**Bước 1** – chạy vòng lặp từ i=0 đến i<n-1 và vòng lặp từ j=0 đến j<n-i-1

**Bước 2** - So sánh phần tử thứ j và phần tử thứ j+1, nếu a[j] > a[j+1] thì hoán đổi a[j] và a[j+1]

**Bước 3**- tăng biến j++ rồi lặp lại bước 2 đến khi mảng được sắp xếp.

Ví dụ step by step:

Sắp xếp dãy số [5 1 4 2 8] này tăng dần.

**Lần lặp đầu tiên:**  
( **5** **1** 4 2 8 ) –> ( **1** **5** 4 2 8 ), Ở đây, thuật toán sẽ so sánh hai phần tử đầu tiên, và đổi chỗ cho nhau do 5 > 1.  
( 1 **5** **4** 2 8 ) –>  ( 1 **4** **5** 2 8 ), Đổi chỗ do 5 > 4  
( 1 4 **5** **2** 8 ) –>  ( 1 4 **2** **5** 8 ), Đổi chỗ do 5 > 2  
( 1 4 2 **5** **8** ) –> ( 1 4 2 **5** **8** ), Ở đây, hai phần tử đang xét đã đúng thứ tự (8 > 5), vậy ta không cần đổi chỗ.

**Lần lặp thứ 2:**  
( **1** **4** 2 5 8 ) –> ( **1** **4** 2 5 8 )  
( 1 **4** **2** 5 8 ) –> ( 1 **2** **4** 5 8 ), Đổi chỗ do 4 > 2  
( 1 2 **4** **5** 8 ) –> ( 1 2 **4** **5** 8 )  
( 1 2 4 **5** **8** ) –>  ( 1 2 4 **5** **8** )  
Bây giờ, dãy số đã được sắp xếp, Nhưng thuật toán của chúng ta không nhận ra điều đó ngay được. Thuật toán sẽ cần thêm một lần lặp nữa để kết luận dãy đã sắp xếp khi và khi khi nó đi từ đầu tới cuối mà không có bất kỳ lần đổi chỗ nào được thực hiện.

**Lần lặp thứ 3:**  
( **1** **2** 4 5 8 ) –> ( **1** **2** 4 5 8 )  
( 1 **2** **4** 5 8 ) –> ( 1 **2** **4** 5 8 )  
( 1 2 **4** **5** 8 ) –> ( 1 2 **4** **5** 8 )  
( 1 2 4 **5** **8** ) –> ( 1 2 4 **5** **8** )

**Flash sort**:

Flash sort là một thuật toán sắp xếp tại chỗ (in-situ, không dùng mảng phụ) có độ phức tạp O(n), không đệ qui, gồm có 3 bước: (1) Phân lớp dữ liệu, tức là dựa trên giả thiết dữ liệu tuân theo 1 phân bố nào đó, chẳng hạn phân bố đều, để tìm 1 công thức ước tính vị trí (lớp) của phần tử sau khi sắp xếp. (2) Hoán vị toàn cục, tức là dời chuyển các phần tử trong mảng về lớp của mình. (3) Sắp xếp cục bộ, tức là để sắp xếp lại các phần tử trong phạm vi của từng lớp.

**Bước 1**: Phân lớp dữ liệu**,** ta có thể chia nhỏ thêm các bước như sau (với a[] là mảng cần được sắp xếp có n phần tử):

* Bước 1: Tìm giá trị nhỏ nhất của các phần tử trong mảng(minVal) và vị trí phần tử  lớn nhất của các phần tử trong mảng(max).
* Bước 2: Khởi tạo 1 vector L có m phần tử
* Bước 3: Đếm số lượng phần tử các lớp theo quy luật, phần tử a[i] sẽ thuộc lớp

k = 1+(m - 1) \* (a[i]-minVal) / (a[max] - minVal).

* Bước 4: Tính vị trí kết thúc của phân lớp thứ j theo công thức L[j] = L[j] + L[j - 1] (j tăng từ 1 đến m - 1).

**Bước 2**:**Hoán vị toàn cục.**

Việc này sẽ hình thành các chu trình hoán vị: mỗi khi ta đem một phẩn tử ở đâu đó đến một vị trí nào đó thì ta phải nhấc phần tử hiện tại đang chiếm chỗ ra, và tiếp tục với phần tử bị nhấc ra và đưa đến chỗ khác cho đến khi quay lại vị trí ban đầu thì hoàn tất vòng lặp.

**Bước 3**: dùng Insertion Sort sắp xếp lại mảng đang bị chia thành các lớp.

Ví dụ step by step:

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generated with low confidence

**Reference:**

// Code & giải thích bubble sort – nguyenvanhieu.vn, *Thuật toán sắp xếp bubble sort minh họa code sử dụng c++* ***,*** available: ***https://nguyenvanhieu.vn/thuat-toan-sap-xep-bubble-sort/***

// Code & giải thích insertion sort – GeeksforGeeks, *Insertion Sort* ***,*** 11 May 2022, available: ***https://www.geeksforgeeks.org/insertion-sort/***

// *Flash sort –* Wikipedia, available: ***https://en.wikipedia.org/wiki/Flashsort***

// Code & giải thích flash sort – codelearn.io, *Thuật toán sắp xếp thần thánh* ***,*** available: ***https://codelearn.io/sharing/flash-sort-thuat-toan-sap-xep-than-thanh***