

# LẬP TRÌNH C CƠ BẢN Sắp xếp – phần 1

#### **NỘI DUNG**

- Sinh dữ liệu thử nghiệm
- Cài đặt 3 thuật toán sắp xếp lựa chọn, chèn, nổi bọt
- Thí nghiệm và đánh giá



## Bài tập 1: sinh dữ liệu cho sắp xếp

- Viết chương trình nhận đầu vào là 3 số nguyên dương n, m, và M, sinh ra n số nguyên dương a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>n</sub> nhận giá trị ngẫu nhiên trong khoảng từ m đến M. Lưu dữ liệu ra file arr-n.txt với định dạng như sau
  - Dòng 1: ghi số nguyên dương n (1 ≤ n ≤ 10<sup>6</sup>)
  - Dòng 2 ghi a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>n</sub>. Các số cách nhau bởi 1 dấu cách

# SẮP XẾP LỰA CHỌN

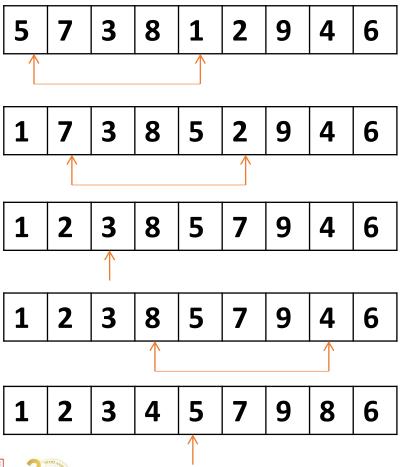
- Chọn số nhỏ nhất xếp vào vị trí thứ 1
- Chọn số nhỏ thứ 2 xếp vào vị trí thứ 2
- Chọn số nhỏ thứ 3 xếp vào vị trí thứ 3

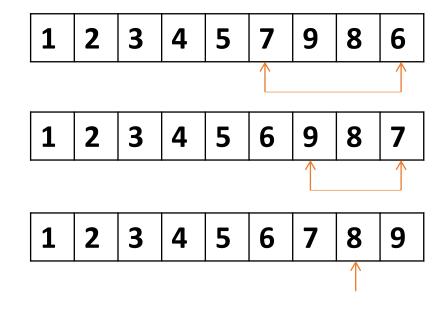
• ...

```
void selectionSort(int A[], int N) {
// index tu 1 -> N
 for(int k = 1; k <= N; k++){
   int min = k;
   for(int j = k+1; j <= N; j++){
     if(A[min] > A[j]) min = j;
   int tmp = A[min];
   A[min] = A[k];
  A[k] = tmp;
```

# SẮP XẾP LỰA CHỌN

• Ví dụ: 5, 7, 3, 8, 1, 2, 9, 4, 6







#### Bài tập 2

- Viết chương trình nhập vào 1 dãy số nguyên dương a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>n</sub>, sắp xếp dãy đã cho theo thứ tự không giảm bằng thuật toán sắp xếp lựa chọn
- Dữ liệu (stdin)
  - Dòng 1: ghi số nguyên dương n (1 ≤ n ≤ 10<sup>6</sup>)
  - Dòng 2: ghi  $a_1, a_2, ..., a_n$ ,  $(1 \le a_i \le 10^6)$
- Kết quả (stdout)
  - Ghi dãy đã được sắp xếp theo thứ tự không giảm, các phần tử cách nhau bởi 1 dấu cách

stdin	stdout		
4	123445		
5 4 3 4 1 2			



#### SẮP XẾP CHÈN

- Thuật toán diễn ra qua các bước lặp k = 2, 3, ..., n
- Tại mỗi bước thứ k: chèn a<sub>k</sub> vào đúng vị trí trong dãy đã được sắp a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>k-1</sub> để thu được dãy được sắp đúng thứ tự
- Sau bước thứ k thì dãy a<sub>1</sub>, a<sub>3</sub>, ..., a<sub>k</sub>
   đã được sắp đúng thứ tự, dãy còn lại a<sub>k+1</sub>, ..., a<sub>n</sub> giữ nguyên vị trí

```
void insertionSort(int A[], int N) {
 // index tu 1 -> N
 for(int k = 2; k <= N; k++){
  int last = A[k];
  int j = k;
  while(j > 1 \&\& A[j-1] >
       last){
    A[j] = A[j-1];
    j--;
 A[j] = last;
```

## SẮP XẾP CHÈN

• Ví dụ: 5, 7, 3, 8, 1, 2, 9, 4, 6

5	7	3	8	1	2	9	4	6
3	5	7	8	1	2	9	4	6
	-	-		-	2			
					2		•	•
						· I	1	· -

1	2	3	5	7	8	9	4	6
1	2	3	4	5	7	8	9	6
1	2	3	4	5	6	7	8	9

#### Bài tập 3

- Viết chương trình nhập vào 1 dãy số nguyên dương  $a_1, a_2, ..., a_n$ , sắp xếp dãy đã cho theo thứ tự không giảm bằng thuật toán sắp xếp chèn
- Dữ liệu (stdin)
  - Dòng 1: ghi số nguyên dương n (1 ≤ n ≤ 10<sup>6</sup>)
  - Dòng 2: ghi  $a_1, a_2, ..., a_n, (1 \le a_i \le 10^6)$
- Kết quả (stdout)
  - Ghi dãy đã được sắp xếp theo thứ tự không giảm, các phần tử cách nhau bởi 1 dấu cách

stdin	stdout
4	123445
5 4 3 4 1 2	



# SẮP XẾP NỔI BỘT

- Duyệt dãy từ trái qua phải (hoặc từ phải qua trái)
  - Tại mỗi bước, so sánh 2 phần tử đứng cạnh nhau và tiến hành đổi chỗ 2 phần tử đó nếu phần tử trước lớn hơn phần tử sau
- Lặp lại quá trình trên khi nào trong dãy vẫn còn 2 phần tử đứng cạnh nhau mà phần tử trước lớn hơn phần tử sau

```
void bubleSort(int A[], int N) {
 // index tu 1 den N
 int swapped;
 do{
   swapped = 0;
   for(int i = 1; i < N; i++){
     if(A[i] > A[i+1]){
       int tmp = A[i];
       A[i] = A[i+1];
       A[i+1] = tmp;
       swapped = 1;
 }while(swapped == 1);
```

# SẮP XẾP NỔI BỌT

• Ví dụ: 5, 7, 3, 8, 1, 2, 9, 4, 6

5 3 7 1 2 8 4 6 9

3 5 1 2 7 4 6 8 9

3 1 2 5 4 6 7 8 9

1 2 3 4 5 6 7 8 9

#### Bài tập 4

- Viết chương trình nhập vào 1 dãy số nguyên dương  $a_1, a_2, ..., a_n$ , sắp xếp dãy đã cho theo thứ tự không giảm bằng thuật toán sắp xếp nổi bọt
- Dữ liệu (stdin)
  - Dòng 1: ghi số nguyên dương n (1 ≤ n ≤ 10<sup>6</sup>)
  - Dòng 2: ghi  $a_1, a_2, ..., a_n$ ,  $(1 \le a_i \le 10^6)$
- Kết quả (stdout)
  - Ghi dãy đã được sắp xếp theo thứ tự không giảm, các phần tử cách nhau bởi 1 dấu cách

stdin	stdout
4	123445
5 4 3 4 1 2	





VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

#### Thank you for your attentions!

