

Môn học: CSD201x_02_VN_ DN Cấu trúc dữ liệu và giải thuật

Học viên: Nguyễn Văn Quyết

Mã học viên: FX10152

I. Kết quả thực nghiệm và phân tích hiệu năng

Thử nghiệm thực tế với 3 trường hợp khác nhau của cùng một mảng a với 15 phần tử:

Array a = {1, 4, 7, 8, 3, 11, 2, 14, 10, 12, 5, 9, 13, 6, 15}

TH1: TỐT NHẤT – Mảng đã được sắp xếp từ đầu.

Array a = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}

- **Bubble Sort**

Thời gian chạy trung bình 12 ms.

Ở trường hợp tốt nhất mảng đã được sắp xếp, thuật toán chỉ duyệt và so sánh n phần tử mà không phải thực hiện hoán vị nên độ phức tạp là $O(n)$

```
Initial array: 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0

STEP 1 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 2 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 3 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 4 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 5 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 6 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 7 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 8 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 9 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 10 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 11 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 12 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 13 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 14 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 15 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0

Running time of bubble sort algorithm: 12 ms
```

■ Selection Sort

Thời gian chạy trung bình 12 ms.

Thuật toán selection sort thực hiện duyệt qua các phần tử, so sánh và thực hiện hoán vị ngay cả khi mảng ban đầu đã được sắp xếp nên độ phức tạp thời gian trung bình là $O(n^2)$

```
Initial array: 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0

STEP 1 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 2 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 3 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 4 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 5 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 6 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 7 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 8 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 9 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 10 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 11 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 12 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 13 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 14 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0

Running time of selection sort algorithm: 12 ms
```

- Insertion Sort

Thời gian chạy trung bình 12 ms.

Mảng đã được sắp xếp nên thuật toán chỉ duyệt và so sánh n phần tử mà không phải thực hiện hoán vị nên độ phức tạp thời gian trung bình là $O(n)$

```
Initial array: 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 1 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 2 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 3 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 4 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 5 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 6 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 7 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 8 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 9 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 10 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 11 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 12 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 13 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 14 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
Running time of insertion sort algorithm: 12 ms
```

TH2: TRUNG BÌNH – Mảng không theo thứ tự.

Array a = {1, 4, 7, 8, 3, 11, 2, 14, 10, 12, 5, 9, 13, 6, 15}

- Bubble Sort

Thời gian chạy trung bình 12 ms.

Trường hợp mảng không theo thứ tự, bubble sort thực hiện duyệt, so sánh các phần tử và hoán đổi vị trí các giá trị không đúng thứ tự, độ phức tạp thuật toán trung bình là $O(n^2)$

```
Initial array: 1.0 4.0 7.0 8.0 3.0 11.0 2.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0

STEP 1 : 1.0 4.0 7.0 3.0 8.0 2.0 11.0 10.0 12.0 5.0 9.0 13.0 6.0 14.0 15.0
STEP 2 : 1.0 4.0 3.0 7.0 2.0 8.0 10.0 11.0 5.0 9.0 12.0 6.0 13.0 14.0 15.0
STEP 3 : 1.0 3.0 4.0 2.0 7.0 8.0 10.0 5.0 9.0 11.0 6.0 12.0 13.0 14.0 15.0
STEP 4 : 1.0 3.0 2.0 4.0 7.0 8.0 5.0 9.0 10.0 6.0 11.0 12.0 13.0 14.0 15.0
STEP 5 : 1.0 2.0 3.0 4.0 7.0 5.0 8.0 9.0 6.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 6 : 1.0 2.0 3.0 4.0 5.0 7.0 8.0 6.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 7 : 1.0 2.0 3.0 4.0 5.0 7.0 6.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 8 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 9 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 10 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 11 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 12 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 13 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 14 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 15 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0

Running time of bubble sort algorithm: 12 ms
```

■ Selection Sort

Thời gian chạy trung bình **12 ms**.

Selection sort thực hiện duyệt, so sánh các phần tử và hoán đổi vị trí các giá trị không đúng thứ tự, độ phức tạp thuật toán trung bình là **$O(n^2)$**

```
Initial array: 1.0 4.0 7.0 8.0 3.0 11.0 2.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0

STEP 1 : 1.0 4.0 7.0 8.0 3.0 11.0 2.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 2 : 1.0 2.0 7.0 8.0 3.0 11.0 4.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 3 : 1.0 2.0 3.0 8.0 7.0 11.0 4.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 4 : 1.0 2.0 3.0 4.0 7.0 11.0 8.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 5 : 1.0 2.0 3.0 4.0 5.0 11.0 8.0 14.0 10.0 12.0 7.0 9.0 13.0 6.0 15.0
STEP 6 : 1.0 2.0 3.0 4.0 5.0 6.0 8.0 14.0 10.0 12.0 7.0 9.0 13.0 11.0 15.0
STEP 7 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 14.0 10.0 12.0 8.0 9.0 13.0 11.0 15.0
STEP 8 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 10.0 12.0 14.0 9.0 13.0 11.0 15.0
STEP 9 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 12.0 14.0 10.0 13.0 11.0 15.0
STEP 10 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 14.0 12.0 13.0 11.0 15.0
STEP 11 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 12 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 13 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 14 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0

Running time of selection sort algorithm: 12 ms
```

- Insertion Sort

Thời gian chạy trung bình 12 ms.

Insertion sort thực hiện duyệt, so sánh các phần tử và hoán đổi vị trí các giá trị không đúng thứ tự, độ phức tạp thời gian trung bình là $O(n^2)$

```
Initial array: 1.0 4.0 7.0 8.0 3.0 11.0 2.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 1 : 1.0 4.0 7.0 8.0 3.0 11.0 2.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 2 : 1.0 4.0 7.0 8.0 3.0 11.0 2.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 3 : 1.0 4.0 7.0 8.0 3.0 11.0 2.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 4 : 1.0 3.0 4.0 7.0 8.0 11.0 2.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 5 : 1.0 3.0 4.0 7.0 8.0 11.0 2.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 6 : 1.0 2.0 3.0 4.0 7.0 8.0 11.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 7 : 1.0 2.0 3.0 4.0 7.0 8.0 11.0 14.0 10.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 8 : 1.0 2.0 3.0 4.0 7.0 8.0 10.0 11.0 14.0 12.0 5.0 9.0 13.0 6.0 15.0
STEP 9 : 1.0 2.0 3.0 4.0 7.0 8.0 10.0 11.0 12.0 14.0 5.0 9.0 13.0 6.0 15.0
STEP 10 : 1.0 2.0 3.0 4.0 5.0 7.0 8.0 10.0 11.0 12.0 14.0 9.0 13.0 6.0 15.0
STEP 11 : 1.0 2.0 3.0 4.0 5.0 7.0 8.0 9.0 10.0 11.0 12.0 14.0 13.0 6.0 15.0
STEP 12 : 1.0 2.0 3.0 4.0 5.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 6.0 15.0
STEP 13 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 14 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
Running time of insertion sort algorithm: 12 ms
```

TH3: XẤU NHẤT – Mảng đã được sắp theo thứ tự ngược lại.

Array a = {15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1}

- Bubble Sort

Thời gian chạy trung bình 12 ms.

Bubble sort thực hiện duyệt, so sánh các phần tử và hoán đổi vị trí các giá trị không đúng thứ tự, độ phức tạp thuật toán trung bình là $O(n^2)$

```
Initial array: 15.0 14.0 13.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0

STEP 1 : 14.0 13.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0 15.0
STEP 2 : 13.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0 14.0 15.0
STEP 3 : 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0 13.0 14.0 15.0
STEP 4 : 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0 12.0 13.0 14.0 15.0
STEP 5 : 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0 11.0 12.0 13.0 14.0 15.0
STEP 6 : 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 7 : 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 8 : 7.0 6.0 5.0 4.0 3.0 2.0 1.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 9 : 6.0 5.0 4.0 3.0 2.0 1.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 10 : 5.0 4.0 3.0 2.0 1.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 11 : 4.0 3.0 2.0 1.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 12 : 3.0 2.0 1.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 13 : 2.0 1.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 14 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 15 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0

Running time of bubble sort algorithm: 12 ms
```

■ Selection Sort

Thời gian chạy trung bình 12 ms.

Selection sort thực hiện duyệt, so sánh các phần tử và hoán đổi vị trí các giá trị không đúng thứ tự, độ phức tạp thuật toán trung bình là $O(n^2)$

```
Initial array: 15.0 14.0 13.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0

STEP 1 : 1.0 14.0 13.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 15.0
STEP 2 : 1.0 2.0 13.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 14.0 15.0
STEP 3 : 1.0 2.0 3.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 13.0 14.0 15.0
STEP 4 : 1.0 2.0 3.0 4.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 12.0 13.0 14.0 15.0
STEP 5 : 1.0 2.0 3.0 4.0 5.0 10.0 9.0 8.0 7.0 6.0 11.0 12.0 13.0 14.0 15.0
STEP 6 : 1.0 2.0 3.0 4.0 5.0 6.0 9.0 8.0 7.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 7 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 8 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 9 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 10 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 11 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 12 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 13 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
STEP 14 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0

Running time of selection sort algorithm: 12 ms
```

- Insertion Sort

Thời gian chạy trung bình 12 ms.

Insertion sort thực hiện duyệt, so sánh các phần tử và hoán đổi vị trí các giá trị không đúng thứ tự, độ phức tạp thuật toán trung bình là $O(n^2)$

```
Initial array: 15.0 14.0 13.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0
STEP 1 : 14.0 15.0 13.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0
STEP 2 : 13.0 14.0 15.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0
STEP 3 : 12.0 13.0 14.0 15.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0
STEP 4 : 11.0 12.0 13.0 14.0 15.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0
STEP 5 : 10.0 11.0 12.0 13.0 14.0 15.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0
STEP 6 : 9.0 10.0 11.0 12.0 13.0 14.0 15.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0
STEP 7 : 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0
STEP 8 : 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 6.0 5.0 4.0 3.0 2.0 1.0
STEP 9 : 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 5.0 4.0 3.0 2.0 1.0
STEP 10 : 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 4.0 3.0 2.0 1.0
STEP 11 : 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 3.0 2.0 1.0
STEP 12 : 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 2.0 1.0
STEP 13 : 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 1.0
STEP 14 : 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
Running time of insertion sort algorithm: 12 ms
```

Nhận xét:

Có thể thấy được qua thực nghiệm, với dữ liệu ít phần tử thì các thuật toán cho thời gian chạy khá tương đồng nhau.

Với mảng đã sắp xếp ngay từ đầu thì **bubble sort** và **insertion sort** sẽ không phải thực hiện hoán vị nên có độ phức tạp là $O(n)$, nhanh hơn so với $O(n^2)$ của **selection sort**. **Selection sort** sẽ luôn thực hiện hoán vị trong tất cả các trường hợp dữ liệu.

Với mảng sắp xếp lộn xộn, không theo thứ tự, cả ba thuật toán đều thực hiện duyệt lần lượt các phần tử, so sánh và hoán vị nên độ phức tạp trung bình là $O(n^2)$. **Bubble sort** và **Insertion sort** có thể nhanh hơn nếu như mảng đã được sắp xếp 1 phần.

Với mảng được sắp xếp ngược với thứ tự cần sắp xếp, đây là trường hợp xấu nhất vì các thuật toán sẽ phải thực hiện số các phép toán tối đa, độ phức tạp thời gian trung bình trong trường hợp này là $O(n^2)$.

Selection sort ở cả ba trường hợp dữ liệu tốt nhất, trung bình và tệ nhất đều có độ phức tạp là như nhau. Như vậy độ phức tạp của selection sort không phụ thuộc vào mảng ban đầu, chỉ phụ thuộc vào số lượng các phần tử. Nguyên nhân là do **selection sort** sẽ luôn thực hiện phép hoán vị trong mọi trường hợp dữ liệu.

II. Công thức tính độ phức tạp thời gian

THUẬT TOÁN	ĐỘ PHỨC TẠP		
	TỐT NHẤT	TRUNG BÌNH	XẤU NHẤT
BUBBLE SORT	$O(n)$	$O(n^2)$	$O(n^2)$
SELECTION SORT	$O(n^2)$	$O(n^2)$	$O(n^2)$
INSERTION SORT	$O(n)$	$O(n^2)$	$O(n^2)$