VISSOFT

# Bài 14
# Thuật toán sắp xếp

Module: ADVANCED PROGRAMMING WITH JAVA

# Mụctiêu

VISSOFT

- Trình bày được thuật toán sắp xếp Nổi bọt
- Cài đặt được thuật toán sắp xếp Nổi bọt
- Trình bày được thuật toán sắp xếp Chọn
- Cài đặt được thuật toán sắp xếp Chọn
- Trình bày được thuật toán sắp xếp Chèn
- Cài đặt được thuật toán sắp xếp Chèn

# Thuật toán sắp xếp

VISSOFT

- Sắp xếp là quá trình xử lý một danh sách các phần tử để đặt chúng theo một thứ tự nào đó dựa trên nội dung thông tin lưu giữ tại mỗi phần tử.
- Dữ liệu được sắp xếp theo thứ tự thường dạng thứ tự số hoặc dạng chữ cái.
- Việc sắp xếp dữ liệu giúp tìm kiếm được tối ưu và dữ liệu dễ đọc hơn.
- Thứ tự sắp xếp có thể là:
  - Thứ tự tăng (increasing order): 1, 3, 4, 6, 8, 9
  - Thứ tự giảm (decreasing order): 9, 8, 6, 4, 3, 1
  - Thứ tự không tăng (non-increasing order):9,8,6,3,3,1
  - Thứ tự không giảm (non-decreasing order):1,3,3,6,8,9

# 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 $a_{ki} > a_{ki-1}$
  - 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. Vậy 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.

# Các thuật toán sắp xếp cơ bản

VISSOFT

- Thuật toán sắp xếp Nổi bọt (Bubble Sort)
- Thuật toán sắp xếp Chọn (Selection Sort)
- Thuật toán sắp xếp Chèn (Insertion Sort)

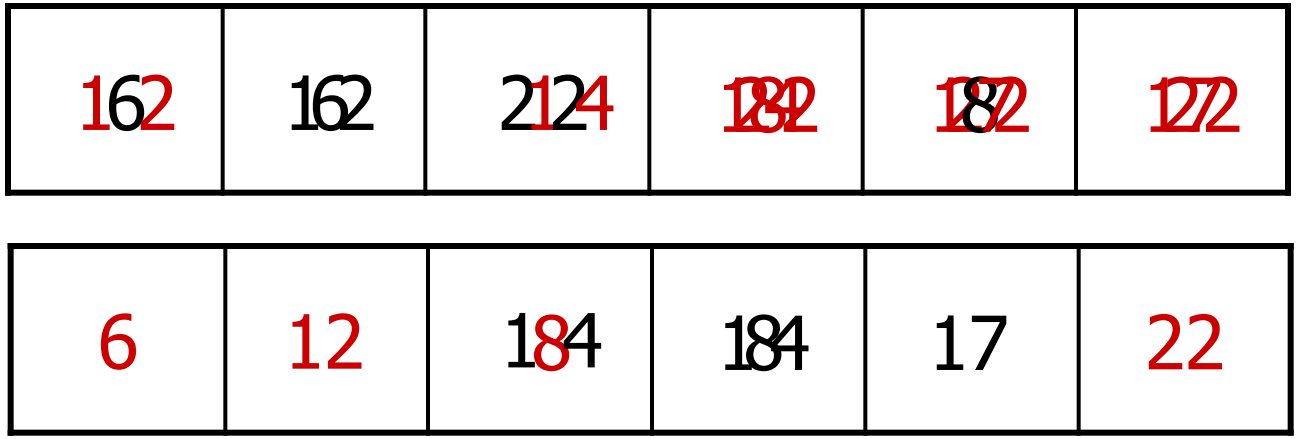
VISSOFT

# Thuật toán sắp xếp nổi bọt

Bubble Sort

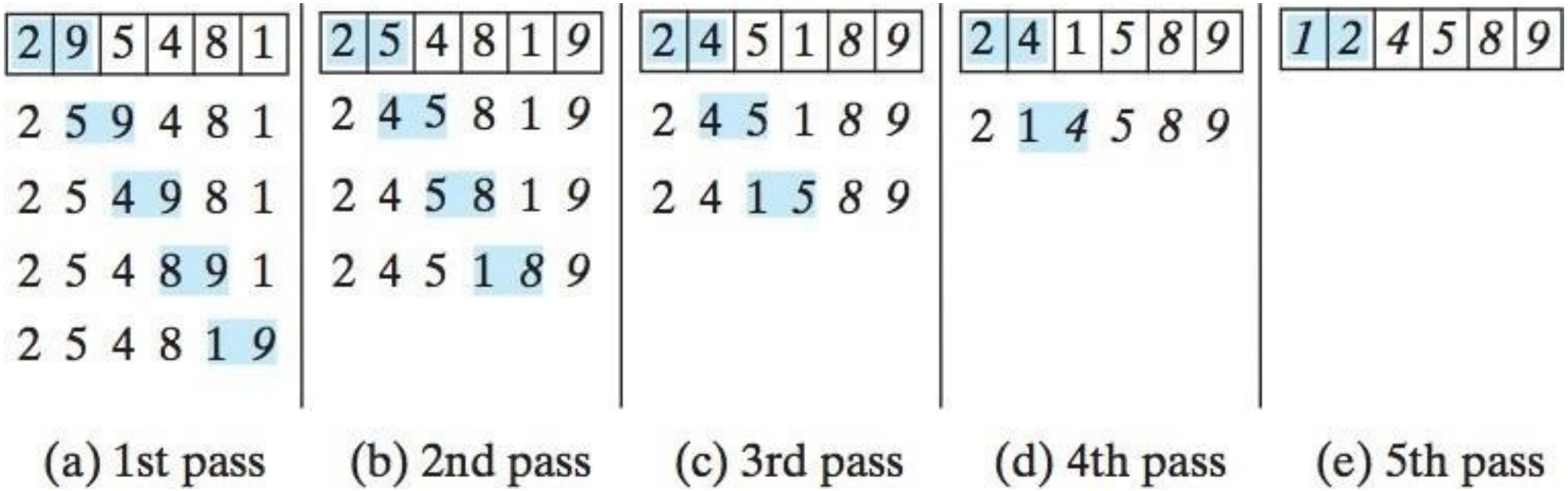
# Sắp xếp nổi bọt (1)

- **Ý tưởng**: Dựa trên việc so sánh cặp phần tử liền kề nhau và tráo đổi thứ tự nếu chúng không theo thứ tự.
- Cho một mảng có **k** phần tử
- Lặp lại các bước sau k-1 lần:
  - Với a[i] và a[i+1]:
  - Nếu a[i] lớn hơn a[i+1] thì đổi vị trí cho nhau.



# Sắp xếp nổi bọt (2)

- Sử dụng bubble sort trên mảng gồm 6 phần tử (2,9,5,4,8,1)



# Sắpxếpnổibọt

- Mô tả thuật toán sắp xếp nổi bọt

```
for (int k = 1; k < list.length; k++) {
  // Perform the kth pass
  for (int i = 0; i < list.length - k; i++) {
    if (list[i] > list[i + 1])
      swap list[i] with list[i + 1];
  }
}
```

***Lưu ý:*** Nếu lệnh if không được thực hiện qua mỗi lần vòng lặp bên ngoài được thực hiện, không cần phải thực hiện lần lặp tiếp theo, bởi vì tất cả các phần tử đã được sắp xếp. Do đó có thể cải tiến đoạn mã lệnh này bằng cách sử dụng biến boolean để kiểm tra.

# Sắpxếpnổibọt

- Cải tiến thuật toán sắp xếp nổi bọt sử dụng biến boolean needNextPass, việc sắp xếp dừng lại khi needNextPass nhận giá trị là false.

```
boolean needNextPass = true;
for (int k = 1; k < list.length && needNextPass; k++) {
  // Array may be sorted and next pass not needed
  needNextPass = false;
  // Perform the kth pass
  for (int i = 0; i < list.length - k; i++) {
    if (list[i] > list[i + 1]) {
      swap list[i] with list[i + 1];
      needNextPass = true; // Next pass still needed
    }
  }
}
```

# Cài đặt thuật toán sắp xếp nổi bọt

VISSOFT

```
public static void bubbleSort(int[] list) {
  boolean needNextPass = true;

  for (int k = 1; k < list.length && needNextPass; k++) {
    // Array may be sorted and next pass not needed
    needNextPass = false;
    for (int i = 0; i < list.length - k; i++) {
      if (list[i] > list[i + 1]) {
        // Swap list[i] with list[i + 1]
        int temp = list[i];
        list[i] = list[i + 1];
        list[i + 1] = temp;

        needNextPass = true; // Next pass still needed
      }
    }
  }
}
```

# Cài đặt thuật toán sắp xếp nổi bọt

VISSOFT

```
int[] list = {2, 3, 2, 5, 6, 1, -2, 3, 14, 12};
bubbleSort(list);
for (int i = 0; i < list.length; i++)
  System.out.print(list[i] + " ");
```

Kết quả

```
-2 1 2 2 3 3 5 6 12 14
```

# Độ phức tạp thuật toán sắp xếp Nổi bọt

VISSOFT

- Trong trường hợp tốt nhất, vòng lặp ngoài thực hiện 1 lần mảng đã được sắp xếp. Số lần so sánh 1 lần thực hiện là n -1. Thời gian thực hiện: O(n)

$$(n-1)+(n-2)+\cdots+2+1$$

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

- Trong trường hợp xấu nhất, vòng lặp ngoài thực hiện n-1 lần. Thời gian thực hiện: O(n2)

VISSOFT

# Thuật toán sắp xếp chọn

Selection Sort

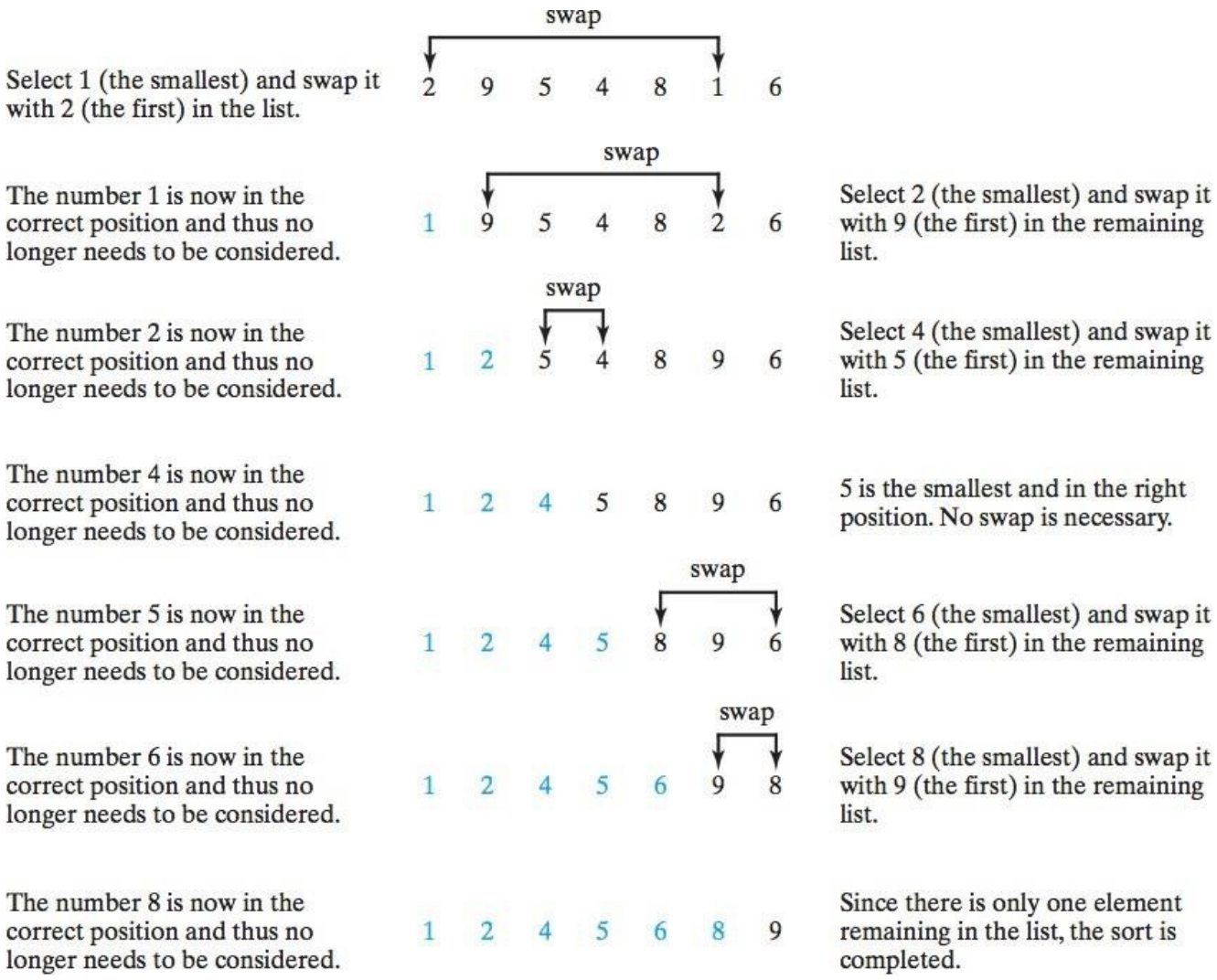
# SắpxếpChọn

VISSOFT

- Ý tưởng:
  - Chọn phần tử nhỏ nhất/lớn nhất trong k 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 k-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.

# SắpxếpChọn

- Cách sắp xếp dãy [2, 9, 5, 4, 8, 1, 6] sử dụng sắp xếp chọn



# Sắp xếp chọn

VISSOFT

- Mô tả thuật toán sắp xếp chọn

```
for (int i = 0; i < list.length - 1; i++) {
  select the smallest element in list[i..list.length-1];
  swap the smallest with list[i], if necessary;
  // list[i] is in its correct position.
  // The next iteration applies on list[i+1..list.length-1]
}
```

# Cài đặt thuật toán sắp xếp chọn

VISSOFT

```
public static void selectionSort(double[] list) {
  for (int i = 0; i < list.length - 1; i++) {
    // Find the minimum in the list[i..list.length-1]
    double currentMin = list[i];
    int currentMinIndex = i;

    for (int j = i + 1; j < list.length; j++) {
      if (currentMin > list[j]) {
        currentMin = list[j];
        currentMinIndex = j;
      }
    }

    // Swap list[i] with list[currentMinIndex] if necessary
    if (currentMinIndex != i) {
      list[currentMinIndex] = list[i];
      list[i] = currentMin;
    }
  }
}
```

VISSOFT

# Thuật toán sắp xếp Chèn

Insertion Sort

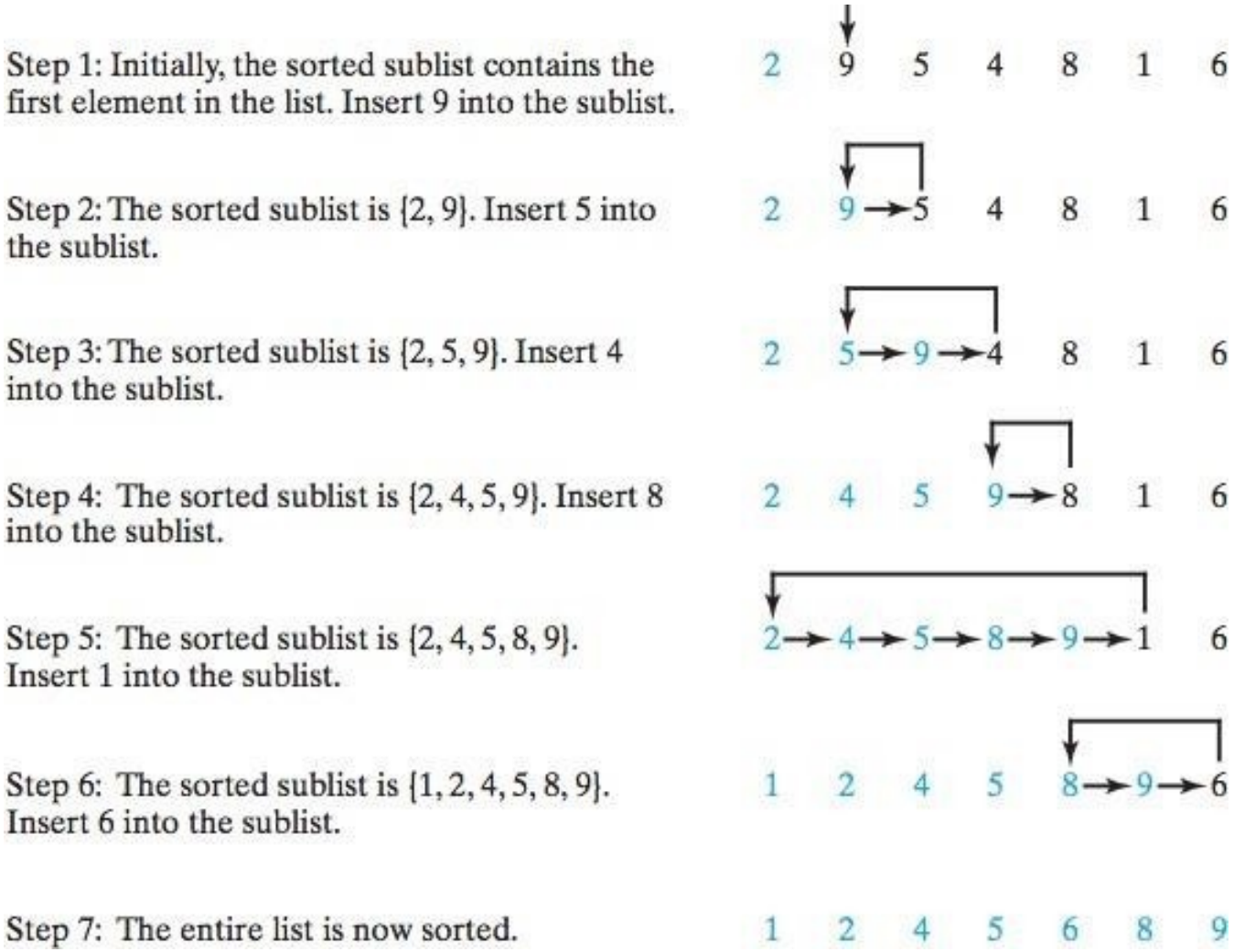
# SắpxếpChèn

VISSOFT

- Ý tưởng:
  - Một danh sách con luôn luôn được duy trì dưới dạng đã qua sắp xếp.
  - Sắp xếp chèn là chèn thêm một phần tử vào danh sách con đã qua sắp xếp. Phần tử được chèn vào vị trí thích hợp sao cho vẫn đảm bảo rằng danh sách con đó vẫn sắp theo thứ tự.

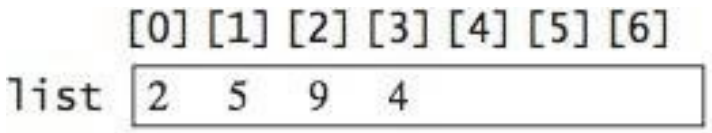
# Sắpxếpchèn

- Sửdụng insertion sort trênmảng gồm 7phần tử [2,9,5,4,8,1,6]



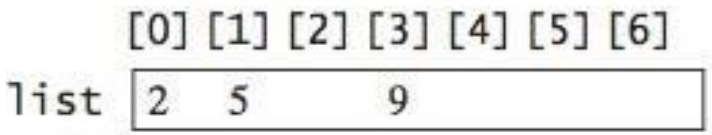
# Sắpxếpchèn

- Phần tử mới được chèn vào danh sách con đã được sắp xếp

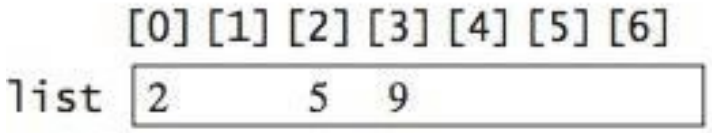



Step 1: Save 4 to a temporary variable `currentElement`

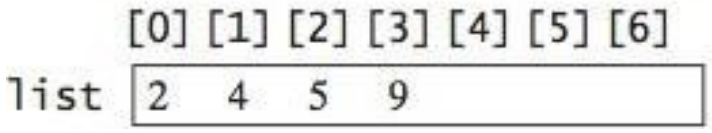
currentElement: 4




Step 2: Move `list[2]` to `list[3]`




Step 3: Move `list[1]` to `list[2]`




Step 4: Assign `currentElement` to `list[1]`

# Sắpxếpchèn

VISSOFT

- Mô tả thuật toán sắp xếp chèn

```
for (int i = 1; i < list.length; i++) {
  insert list[i] into a sorted sublist list[0..i-1] so that
  list[0..i] is sorted.
}
```

# Cài đặt thuật toán – Sắp xếp chèn

VISSOFT

```
public void insertionSort(int []list){
    for (int i = 1; i < list.length; i++){
        //insert list[i] into a sorted sublist list[0...i-1]
        //so that list[0..i] is sorted
        int currentElement = list[i];
        int k;
        for(k = i - 1; k >=0 && list[k] > currentElement; k--
            ){ list[k+1] = list[k];
        }
        //insert the current elemtn into list[k+1]
        list[k+1] = currentElement;
    }
}
```

# Độ phức tạp thuật toán sắp xếp Nổi bọt

VISSOFT

- Thời gian thực hiện: O(n2)

$$
\begin{aligned}
T(n) &= (2 + c) + (2 \times 2 + c) + \cdots + (2 \times (n - 1) + c) \\
&= 2(1 + 2 + \cdots + n - 1) + c(n - 1) \\
&= 2\frac{(n - 1)n}{2} + cn - c = n^2 - n + cn - c \\
&= O(n^2)
\end{aligned}
$$

# [Thực hành] Cài đặt thuật toán sắp xếp nổi bọt

VISSOFT

# [Thực hành] Cài đặt thuật toán sắp xếp chọn

VISSOFT

# [Thực hành] Minh hoạ thuật toán sắp xếp chèn

VISSOFT

# [Bài tập] Cài đặt thuật toán sắp xếp chèn

VISSOFT

# [Bài tập] Minh hoạ thuật toán sắp xếp chèn

VISSOFT

# [Bàitập]Viếtcácphươngthứckiểmtratrậttựsắpxếpcủamảng

VISSOFT

# Tổng kết

VISSOFT

- Sắp xếp nổi bọt dựa trên việc so sánh cặp phần tử liền kề nhau và tráo đổi thứ tự nếu chúng không theo thứ tự.
- Sắp xếp chọn là chọn phần tử nhỏ nhất trong k phần tử ban đầu, đưa phần tử này về vị trí đầu dãy hiện tại, sau đó loại nó khỏi danh sách sắp xếp tiếptheo.
- Sắp xếp chèn là chèn thêm một phần tử vào danh sách con đã qua sắp xếp. Phần tử được chèn vào vị trí thích hợp sao cho vẫn đảm bảo rằng danh sách con đó vẫn sắp theo thứ tự.

VISSOFT

# Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: Debug