

Bài 8.3: Thuật toán sắp xếp chọn

- ✓ Các đặc điểm
- ✓ Thuật toán selection sort
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Các đặc điểm

- Là một thuật toán sắp xếp tập hợp có độ phức tạp $O(n^2)$.
- Thuật toán thực hiện việc sắp xếp bằng cách lần lượt tìm các phần tử nhỏ nhất từ phần chưa được sắp xếp đưa về phần đầu tập hợp.
- Thuật toán chia tập hợp(mảng) cần sắp xếp thành 2 phần:
 - Phần đầu mảng chứa các phần tử đã được sắp xếp.
 - Phần sau của mảng chứa các phần tử chưa được sắp xếp.
- Tại mỗi lần lặp, phần tử nhỏ nhất trong phần chưa được sắp xếp sẽ được đưa sang đúng vị trí của nó ở phần mảng đã được sắp xếp.

Thuật toán sắp xếp chọn

➤ Mã giả của thuật toán selection sort:

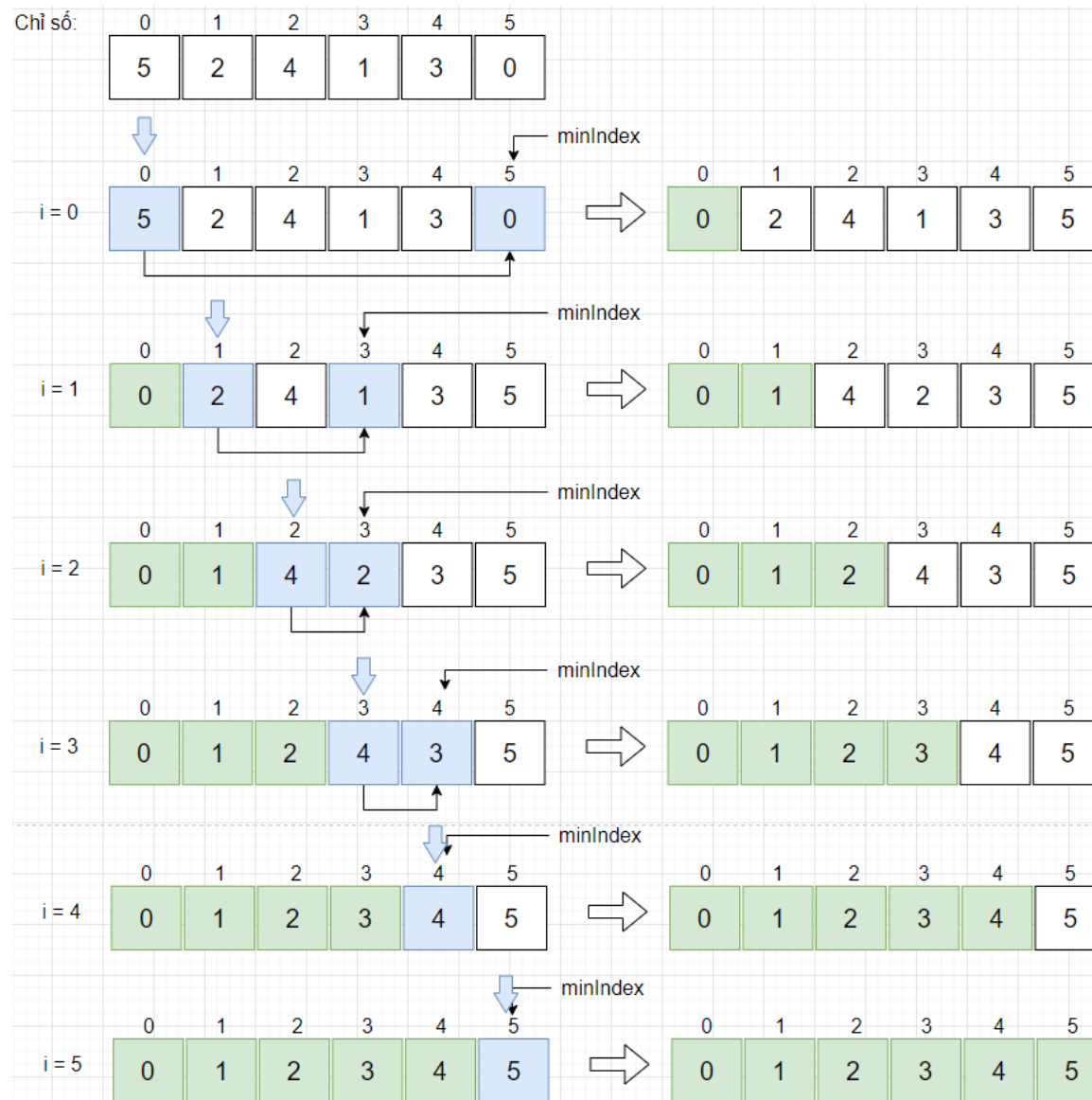
```
void selectionSort(arr[]) { // mảng đầu vào
    n = arr.length // lấy số phần tử của mảng
    for (i from 0 to n - 2) { // cho i từ 0 đến n-2
        minIndex = i; // giả định chỉ số phần tử min là i
        for (j from i + 1 to n - 1) { // tìm chỉ số phần tử min
            if (arr[j] < arr[minIndex]) {
                minIndex = j; // cập nhật chỉ số phần tử min
            }
        }
        // tiến hành đổi chỗ hai phần tử tại vị trí i và minIndex
        tmp = arr[i];
        arr[i] = arr[minIndex];
        arr[minIndex] = tmp;
    }
}
```

Thuật toán sắp xếp chọn

➤ Mã thật của thuật toán selection sort:

```
public static <T extends Comparable<T>> void selectionSort(T[] arr) {  
    int n = arr.length;  
    for (int i = 0; i < n - 1; i++) {  
        int minIndex = i;  
        for (int j = i + 1; j < n; j++) {  
            if (arr[j].compareTo(arr[minIndex]) < 0) {  
                minIndex = j;  
            }  
        }  
        T tmp = arr[i];  
        arr[i] = arr[minIndex];  
        arr[minIndex] = tmp;  
    }  
}
```

Minh họa thuật toán



Nội dung tiếp theo

Thuật toán sắp xếp insertion sort