Bài 5.3: Thuật toán quicksort

- ✓ Các đặc điểm thuật toán
- ✓ Mã giả và triển khai
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Các đặc điểm thuật toán

- ✓Thuật toán sắp xếp nhanh có độ phức tạp O(nlogn).
- ✓Đây là thuật toán thường sử dụng trong các chức năng sắp xếp tập hợp và trong phỏng vấn tuyển dụng.
- ✓Hiệu năng của một thuật toán phụ thuộc vào nhiều yếu tố bao gồm cả vấn đề cần giải quyết và bộ dữ liệu đầu vào để thuật toán vận hành.
- ✓Quicksort là thuật toán thuộc dạng chia để trị. Nó chọn 1 phần tử làm mốc và chia tập hợp cần sắp xếp thành 3 phần:
  - ✓ Phần 1: chứa các phần tử nhỏ hơn hoặc bằng phần tử chọn làm mốc.
  - ✓ Phần 2: chỉ gồm phần tử chọn làm mốc.
  - ✓ Phần 3: gồm các phần tử lớn hơn phần tử chọn làm mốc.
- ✓Có nhiều cách chọn phần tử làm mốc: ở đầu, cuối, giữa hoặc ngẫu nhiên trong tập hợp.
- ✓Thuật toán minh họa trong bài học này chọn phần tử ở cuối tập hợp làm mốc.

Mã giả của thuật toán

✓Sau đây là mã giả của thuật toán quicksort trên mảng:

```
// thuật toán quicksort:
void quicksort(arr[], leftIndex, rightIndex) {
    if (leftIndex < rightIndex) { // nếu biên trái < biên phải
        p = partition(arr, leftIndex, rightIndex)
        quicksort(arr, leftIndex, p - 1)
        quicksort(arr, p + 1, rightIndex)
    }
}

// thuật toán phân mảnh và sắp xếp các phần tử
int partition(arr[], left, right) {
    pivot = arr[right] // giá trị phần tử được chọn làm mốc
    i = left // khởi tạo biến chạy i bắt đầu từ biên trái
    for (j from left to right) { // chạy j từ biên trái đến biên phải
        if (arr[j] < pivot) {
            tmp = arr[i];
            arr[i] = arr[j];
            arr[j] = tmp;
            i = i + 1
        }
    }
    swap(arr[i], arr[right]) // đổi chỗ arr[i] và arr[right]
    return i // trả về vị trí phần tử chọn làm mốc kế tiếp
}
```

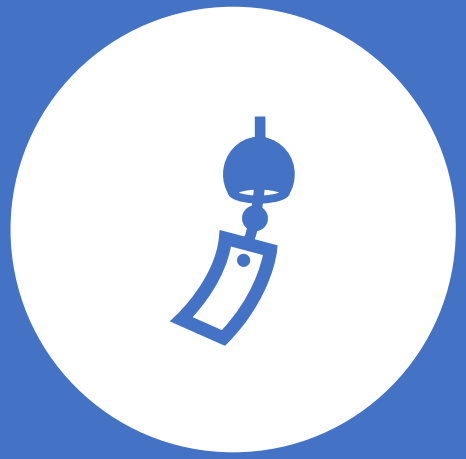
Triển khai của thuật toán

```
// Phương thức sắp xếp nhanh
3 references
static void QuickSort(int[] arr, int leftIndex, int rightIndex)
{
    if (leftIndex < rightIndex)
    {
        int p = Partition(arr, leftIndex, rightIndex);
        QuickSort(arr, leftIndex, p - 1);
        QuickSort(arr, p + 1, rightIndex);
    }
}

// Phương thức phân mảnh và trả về vị trí phần tử mốc
1 reference
static int Partition(int[] arr, int left, int right)
{
    int pivot = arr[right];
    int i = left;
    for (int j = left; j <= right; j++)
    {
        if (arr[j] < pivot)
        {
            int tmp = arr[i];
            arr[i] = arr[j];
            arr[j] = tmp;
            i++;
        }
    }
    Swap(ref arr[i], ref arr[right]);
    return i;
}
```

```
// Phương thức tráo đổi giá trị của hai phần tử a, b
1 reference
static void Swap(ref int a, ref int b)
{
    int tmp = a;
    a = b;
    b = tmp;
}

// Phương thức hiển thị các phần tử trong mảng
2 references
static void ShowArray(int[] arr)
{
    foreach (var item in arr)
    {
        Console.Write(item + " ");
    }
    Console.WriteLine();
}
```



Nội dung tiếp theo

Thuật toán tìm kiếm nhanh