

Bài 9.7: Thuật toán sắp xếp nhanh

- ✓ Các đặc điểm
- ✓ Thuật toán sắp xếp nhanh
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Các đặc điểm

- Thuật toán sắp xếp nhanh có độ phức tạp $O(n \log n)$.
- Đây là thuật toán thường được áp dụng trong các chức năng sắp xếp.
- Tuy vậy, hiệu quả của một thuật toán còn phụ thuộc vào vấn đề cần giải quyết và bộ dữ liệu đầu vào mà nó vận hành.
- Quicksort là một thuật toán thuộc dạng chia để trị. Nó chọn một phần tử làm mốc và chia tập hợp cần sắp xếp thành 3 phần:
 - Phần 1: chứa các phần tử nhỏ hơn phần tử chọn làm mốc.
 - Phần 2: 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 cuối tập hợp làm mốc.

Thuật toán sắp xếp nhanh

➤ Mã giả của thuật toán quick sort:

```
// thuật toán quicksort
// arr: mảng chứa các phần tử đầu vào
// leftIndex: chỉ số biên trái
// rightIndex: chỉ số biên phải
function quicksort(arr[], leftIndex, rightIndex):
    if(leftIndex < rightIndex):
        p = partition(arr, leftIndex, rightIndex)
        quicksort(arr, leftIndex, p - 1)
        quicksort(arr, p + 1, rightIndex)

// thuật toán phân mảng và sắp xếp các phần tử
// arr: mảng chứa các phần tử đầu vào
// leftIndex: chỉ số biên trái
// rightIndex: chỉ số biên phải
function partition(arr[], left, right):
    pivot = arr[right] // phần tử chọn làm mốc
    i = left // bắt đầu từ biên trái
    for(j từ left tới right):
        if(arr[j] < pivot): // phần tử tại vị trí j nhỏ hơn mốc
            swap(arr[i], arr[j]) // trao đổi
            i++ // tăng i
    swap(arr[i], arr[right]) // trao đổi phần tử tại vị trí i và right
    return i // trả về vị trí phần tử mốc kế tiếp
```

Thuật toán sắp xếp nhanh

➤ Mã thật của thuật toán quick sort:

```
// thuật toán phân mảnh và sắp xếp các phần tử
template<class T> int partition(T* arr, int left, int right) {
    T pivot = arr[right];
    int i = left;
    for (int j = left; j <= right; j++)
    {
        if (arr[j] < pivot) {
            swap(arr[i], arr[j]);
            i++;
        }
    }
    swap(arr[i], arr[right]);
    return i;
}

// thuật toán sắp xếp nhanh
template<class T> void quickSort(T* arr, int left, int right) {
    if (left < right) {
        int p = partition(arr, left, right);
        quickSort(arr, left, p - 1);
        quickSort(arr, p + 1, right);
    }
}
```

Nội dung tiếp theo

Thuật toán sắp xếp đếm