

BÀI 9

LẬP TRÌNH THUẬT TOÁN SẮP XẾP NHANH

Học xong bài này, em sẽ:

- ✓ Hiểu được ý tưởng của thuật toán sắp xếp nhanh.
- ✓ Viết được chương trình thực hiện sắp xếp nhanh một dãy số dựa trên các mã lệnh thuật toán phân đoạn cho trước.



Nếu cần chọn một trong hai việc sau đây, em sẽ chọn làm việc nào? Vì sao?

- 1) Từ mô tả thuật toán bằng liệt kê các bước, viết chương trình Python thực hiện thuật toán.
- 2) Từ chương trình Python thực hiện thuật toán, viết lại ngắn gọn ý tưởng chính của thuật toán.

1. Lược đồ phân đoạn trong sắp xếp nhanh

Thuật toán sắp xếp nhanh (Quick Sort)

Thuật toán theo chiến lược *chia để trị*, lặp lại nhiều lần việc *phân đoạn* dãy đầu vào thành hai đoạn con.

Sau một lần phân đoạn, chỉ cần sắp xếp trong nội bộ hai đoạn con. Bài toán sắp xếp ban đầu được chia thành hai bài toán con nhỏ hơn. Việc phân đoạn lặp lại nhiều lần. Hai đoạn con lại được tiếp tục phân đoạn thành 4 đoạn con, 8 đoạn con,... cho đến khi tất cả các đoạn con đều chỉ còn không quá một phần tử. Dãy ban đầu được sắp xếp xong.

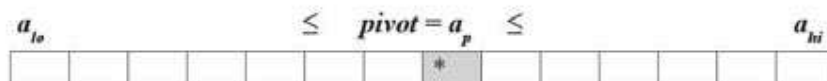
Lược đồ phân đoạn dãy số

– Lấy giá trị của một phần tử trong dãy làm pivot (giá trị chốt). Giá trị pivot có thể là bất cứ phần tử nào trong dãy. Phần tử này cũng được gọi là pivot cho ngắn gọn.

– Kết quả phân đoạn (*Hình 1*): đoạn con ở nửa dãy bên trái chỉ gồm các phần tử nhỏ hơn hay bằng pivot; đoạn con nửa dãy bên phải chỉ gồm các phần tử lớn hơn hay bằng pivot, phần tử làm pivot được chuyển đến đúng vị trí phân tách hai đoạn.

– Hàm thực hiện phân đoạn cần trả về vị trí phân tách dãy thành hai đoạn con vì sau đó sẽ sắp xếp chỉ trong nội bộ hai đoạn con.

Sau khi phân đoạn, chỉ cần sắp xếp trong nội bộ hai đoạn con thì hoàn thành việc sắp xếp cả dãy số. Bài toán sắp xếp ban đầu được chia thành hai bài toán con nhỏ hơn.



Hình 1. Kết quả phân đoạn: đoạn trái = $\{i \mid l_0 \leq i \leq p - 1\}$, đoạn phải = $\{j \mid p + 1 \leq j \leq h_i\}$

Có những lược đồ phân đoạn khác nhau để đạt mục đích trên.

2 Thuật toán sắp xếp nhanh áp dụng phân đoạn Lomuto



Em hãy cho biết lược đồ phân đoạn Lomuto theo mã giả cho trong Hình 2 có đáp ứng yêu cầu phân đoạn để sắp xếp nhanh như trình bày ở mục 1 hay không.

Hình 2 là mã giả của thuật toán Lomuto phân đoạn dãy số a ; lo là chỉ số đầu mút trái; hi là chỉ số đầu mút phải.

```
pivot ← a[hi]           # pivot là phần tử cuối dãy số
i ← lo - 1              # i là vị trí phân tách
for j in {j | lo ≤ j ≤ hi}:
    if a[j] ≤ pivot:     # kiểm tra yêu cầu phân đoạn
        i ← i + 1       # tiến i một vị trí
        swap(a[i], a[j]) # đổi chỗ a[j] về vị trí phân tách
return i                # trả về vị trí phân tách
```

Hình 2. Mã giả thực hiện phân đoạn Lomuto

Ý tưởng thuật toán Lomuto: Chọn pivot là giá trị phần tử đứng cuối dãy số. Duy trì chỉ số i ở vị trí phân tách; duyệt dãy số bằng một chỉ số j khác và đảo giá trị các phần tử sao cho các phần tử ở vị trí từ $i - 1$ về đầu mút trái nhỏ hơn hay bằng pivot; các phần tử từ vị trí $i + 1$ đến j lớn hơn pivot, riêng phần tử ở vị trí i đúng bằng pivot.

Hình 3 là mã lệnh Python thực hiện sắp xếp nhanh bằng phân đoạn Lomuto.

```
File Edit Format Run Options Window Help
1 def phandoanLomuto(a, lo, hi):
2     i = (lo-1)           #i là vị trí phân tách
3     pivot = a[hi]
4     for j in range(lo, hi): #Duyệt dãy a[lo...hi]
5         if a[j] ≤ pivot:   #Phần tử a[j] ≤ pivot
6             i = i+1       #Tăng i lên,
7             a[i], a[j] = a[j], a[i] #Đảo giá trị a[i], a[j]
8     a[i+1], a[hi] = a[hi], a[i+1] #Đảo giá trị a[i+1], a[hi]
9     return (i+1)         #Hết vòng lặp, i là vị trí phân đoạn,
                             #a[i] = pivot
11 def quickSort(a, lo, hi):
12     if lo < hi:
13         p = phandoanLomuto(a, lo, hi) #p là vị trí phân tách
14         #a[p] đã ở đúng chỗ
15         quickSort(a, lo, p-1)         #Sắp xếp đoạn bên trái
16         quickSort(a, p+1, hi)         #Sắp xếp đoạn bên phải
```

Hình 3. Mã lệnh Python thuật toán sắp xếp nhanh sử dụng phân đoạn Lomuto

3 Thuật toán sắp xếp nhanh áp dụng phân đoạn Hoare

Lược đồ phân đoạn Hoare

Hoare là tác giả của thuật toán sắp xếp nhanh. Ý tưởng chính của thuật toán là đổi chỗ nhảy qua điểm phân tách (pivot), rà soát từ hai phía, trái và phải, cùng tiến dần từng bước vào giữa. Tạm dừng khi phát hiện phần tử vi phạm yêu cầu phân đoạn ở mỗi phía và đổi chỗ chúng cho nhau. Rà soát từ hai điểm tạm dừng đi vào giữa cho đến khi gặp nhau thì kết thúc việc phân đoạn. Điểm gặp nhau là vị trí phân tích dãy thành hai đoạn con.

Hình 4 là mã giả của thuật toán sắp xếp nhanh áp dụng phân đoạn Hoare. Hình 5 là mã lệnh Python của thuật toán phân đoạn dãy số a , xuất phát với pivot là đầu dãy.

Bước 1. $\text{pivot} \leftarrow a[lo]; i \leftarrow lo; j \leftarrow hi$;
Bước 2.
 Tìm từ i qua phải đến khi $a[i] \geq \text{pivot}$;
 Tìm từ j qua trái đến khi $a[j] \leq \text{pivot}$;
Bước 3.
 if $i < j$:
 Hoán đổi a_i với a_j ; cho i qua phải;
 cho j qua trái; đến Bước 2;
 else:
 Trả về j để tiếp tục phân đoạn 2 dãy con:
 $a[lo] \dots a[j]$ và $a[j+1] \dots a[hi]$
Kết thúc;

Hình 4. Mã giả thực hiện phân đoạn Hoare

```
File Edit Format Run Options Window Help
1 def partitionHoare(a, lo, hi):
2     pivot = a[lo]
3     i, j = lo, hi
4     phanDoan = True
5     while phanDoan:                                #Đang phân đoạn
6         #i qua phải đến khi a[i] >= pivot
7         while a[i] < pivot:
8             i = i + 1
9         #j qua trái đến khi a[j] <= pivot
10        while a[j] > pivot:
11            j = j - 1
12        if i < j:                                    #i chưa gặp j
13            #Hoán đổi a[i] với a[j]
14            a[i], a[j] = a[j], a[i]
15            i = i + 1                                #i qua phải
16            j = j - 1                                #j qua trái
17        else:
18            phanDoan = False                          #Kết thúc phân đoạn
19        return j
20 def quickSortHoare(a, lo, hi):
21     if lo < hi:
22         p = partitionHoare(a, lo, hi)
23         quickSortHoare(a, lo, p)
24         quickSortHoare(a, p+1, hi)
```

Hình 5. Mã lệnh thực hiện phân đoạn Hoare

Thực hành

Nhiệm vụ 1. Viết chương trình thực hiện sắp xếp nhanh một dãy số và chạy thử kiểm tra.

a) Dựa trên mã lệnh thuật toán cho trong *Hình 3*.

b) Dựa trên mã lệnh thuật toán cho trong *Hình 5*.

Nhiệm vụ 2. Bổ sung thêm các câu lệnh in kết quả trung gian vào các chương trình nói trên để có thể quan sát diễn biến từng bước thực hiện sắp xếp nhanh một dãy số.



Em hãy thực hiện các công việc sau:

a) Sửa lại thủ tục phân đoạn để có hàm `quickSort_down` sắp xếp theo thứ tự giảm dần.

Gợi ý: Sửa đổi phép so sánh trong câu lệnh `if a[j] <= pivot:` thành `if a[j] >= pivot:`

b) Tiếp tục sửa lại để có hàm `quickSort_tuple_down` sắp xếp danh sách các cặp, ví dụ (*tên học sinh, điểm môn học*) theo điểm môn học giảm dần.

Gợi ý: Sửa đổi đầu vào thành danh sách các cặp (*tên học sinh, điểm môn học*) và thực hiện so sánh theo điểm môn học.



Câu 1. Em hãy giải thích tại sao lại nói thuật toán sắp xếp nhanh (QuickSort) theo chiến lược “chia để trị”.

Câu 2. Theo em thì diễn biến từng bước sắp xếp nhanh một dãy số cụ thể dùng phân đoạn Lomuto sẽ giống hay sẽ khác với dùng phân đoạn Hoare?

Tóm tắt bài học

- ✔ Thuật toán sắp xếp nhanh có thể áp dụng một trong hai lược đồ phân đoạn: theo Lomuto hoặc theo Hoare.
- ✔ Lược đồ Lomuto thực hiện phân đoạn bằng cách kiểm tra theo một chiều từ trái sang phải, đổi chỗ và dịch chuyển dần vị trí phân tách hai dãy con cho đến khi thỏa mãn yêu cầu phân đoạn.
- ✔ Lược đồ Hoare thực hiện phân đoạn bằng cách kiểm tra theo hai chiều, từ hai đầu dãy số tiến dần vào giữa, đổi chỗ để thỏa mãn yêu cầu phân đoạn; kết thúc khi gặp nhau.