

CHƯƠNG 3: QUICK SORT

GVGD: Ths NGUYỄN CHÍ THANH

Email: thanh.nc@ou.edu.com

NỘI DUNG

1. Ý tưởng

2. Thuật toán

3. Ví dụ

4. Cài đặt

5. Độ phức tạp

1. PHÁT BIỂU BÀI TOÁN



Cho một tập các số nguyên gồm n phần tử:


$$a_0, a_2, a_3, \dots, a_{n-1}$$


Yêu cầu: hãy thực hiện sắp xếp n phần tử này theo thứ tự tăng dần như sau:

$$a_0, a_2, a_3, \dots, a_{n-1}$$

Với $a_0 \leq a_2 \leq a_3 \leq \dots \leq a_{n-1}$

MÔ HÌNH HÓA BÀI TOÁN

 **Đầu vào:** một danh sách đặc (các số nguyên) gồm có n phần tử $a_0, a_2, a_3, \dots, a_{n-1}$.

 **Đầu ra:** một danh sách đặc (các số nguyên) gồm có n phần tử: $a_0, a_2, a_3, \dots, a_{n-1}$ ($a_0 \leq a_2 \leq a_3 \leq \dots \leq a_{n-1}$)

BIỂU DIỄN BÀI TOÁN TRÊN MÁY TÍNH

```
# define MAX 100
```

```
int a[MAX];
```

```
int n; // n là tổng số phần tử hiện có trong danh sách,  $0 \leq n < \text{MAX}$ 
```

Ý TƯỞNG QUICK SORT

- Với một danh sách đặc a , có n phần tử từ $a[0]$ đến $a[n-1]$ như sau:
 $a[0], a[1], a[2], a[3], \dots, a[n-1]$

Phần tử:	$a[0]$	$a[1]$	$a[2]$	$a[3]$	\dots	\dots	$a[n-1]$
Vị trí:	0	1	2	3	\dots	\dots	$n-1$

Ý TƯỞNG QUICK SORT

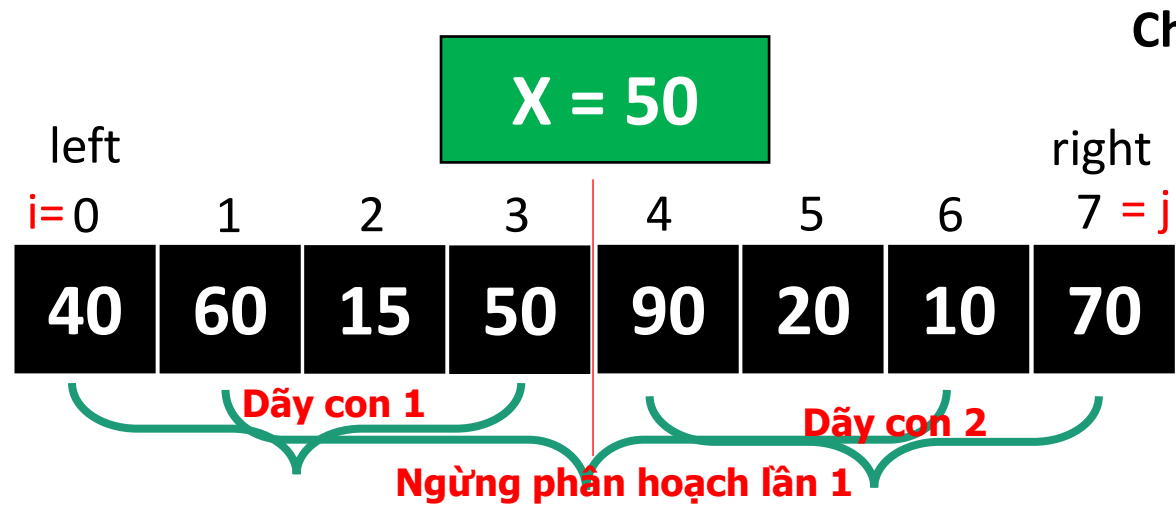
- Bước 1: Phân hoạch các phần tử trong danh sách làm 3 đoạn

Đoạn 1: Chứa các phần tử nhỏ hơn hoặc bằng x ($a[k] \leq x, k=0..j$)

Đoạn 2: Chứa các phần tử lớn hơn hoặc bằng x ($a[k] \geq x, k=i..n-1$)

$a[k] \leq x$	$a[h] \leq x$	$a[g] \geq x$
---------------	---------------	---------------

- Sau khi phân hoạch danh sách a thành 2 đoạn, ta tiếp tục làm động tác phân hoạch cho đoạn 1, và đoạn 2. Quá trình phân hoạch này dừng khi danh sách cần phân hoạch chỉ còn 1 phần tử.



- Ta được dãy con 1 gồm: $a[0], a[1], a[2], a[3]$.
- Ta được dãy con 2 gồm: $a[4], a[5], a[6], a[7]$.
- Trên dãy con 1, tiếp tục phân hoạch với $\text{left} = 0, \text{right} = j = 3$;
- Trên dãy con 2, tiếp tục phân hoạch với $\text{left} = i = 4, \text{right} = 7$;
- Tiếp tục cho đến khi dãy con chỉ có 1 phần tử. Khi đó danh sách sẽ được sắp xếp

THUẬT GIẢI QUICK SORT

```
void QuickSort(int a[], int left, int right)
{
    int x = a[(left+right)/2];
    int i=left;
    int j=right;
    while(i<j)
    {
        while(a[i]<x) i++;
        while(a[j]>x) j--;
```

```
        while(a[j]>x) j--;
        if(i<=j){
            int temp=a[i];
            a[i]=a[j];
            a[j]=temp;
            i++; j--;
        }
    }
    if(left<j) QuickSort(a,left,j);
    if(i<right) QuickSort(a,i,right);
}
```

ĐÁNH GIÁ ĐỘ PHỨC TẠP

TRƯỜNG HỢP TỐT NHẤT

40	60	15	50	90	20	10	70
----	----	----	----	----	----	----	----

Chọn được giá trị X mà *sau khi phân hoạch* theo X ta được 2 đoạn có độ dài lệch nhau không quá 1.

X = 50								
40	10	15	20		90	50	60	70

TRƯỜNG HỢP XẤU NHẤT

40	60	15	50	90	20	10	70
----	----	----	----	----	----	----	----

Chọn được giá trị X mà *sau khi phân hoạch* theo X ta được 2 đoạn có độ dài lệch nhau $= n-4$

40	60	15	50	10	20		90	70
----	----	----	----	----	----	--	----	----

$X = 70$

TÍNH ĐỘ PHỨC TẠP THUẬT TOÁN QUICKSORT

```
void QuickSort(int a[], int left, int right)
{
    int x = a[(left+right)/2];
    int i=left;
    int j=right;
    while(i<j)
    {
        while(a[i]<x) i++;
        while(a[j]>x) j--;
```

```
        while(a[j]>x) j--;
        if(i<=j){
            int temp=a[i];
            a[i]=a[j];
            a[j]=temp;
            i++; j--;
        }
    }
    if(left<j) QuickSort(a,left,j);
    if(i<right) QuickSort(a,i,right);
}
```

XÁC ĐỊNH CÔNG THỨC ĐỆ QUY

```

void QuickSort(int a[], int left, int right)
{
    int x = a[(left+right)/2];
    int i=left;
    int j=right;
    while(i<j)
    {
        while(a[i]<x) i++;
        while(a[j]>x) j--;
        while(a[j]>x) j--;
        if(i<=j){
            int temp=a[i];
            a[i]=a[j];
            a[j]=temp;
            i++; j--;
        }
    }
    if(left<j) QuickSort(a,left,j);
    if(i<right) QuickSort(a,i,right);
}

```

12/09/2018

KHOA CÔNG NGHỆ THÔNG TIN

$$\begin{cases} T(n) = C_1 \text{ (khi } n = 1) \\ T(n) = 2T\left(\frac{n}{2}\right) + C_2n \text{ (} n > 1) \end{cases}$$

$$\begin{aligned} T(n) &= 2\left[2T\left(\frac{n}{4}\right) + C_2\frac{n}{2}\right] + C_2n \\ &= 4T\frac{n}{4} + 2C_2n = 4\left[2T\left(\frac{n}{8}\right) + C_2\frac{n}{4}\right] + 2C_2n \\ &= 4T\left(\frac{n}{4}\right) + 2C_2n = 8T\left(\frac{n}{8}\right) + 3C_2n \end{aligned}$$

$$\dots = 2^k T\left(\frac{n}{2^k}\right) + kC_2n$$

Chương trình dừng khi $\frac{n}{2^k} = n = 2^k \Rightarrow k = \log_2 n$

$$\Leftrightarrow 2^{\log_2 n} T(1) + \log_2 n C_2n = 2^{\log_2 n} C_1 + \log_2 n C_2n$$

$$T(n) = 2^{\log_2 n} C_1 + \log_2 n C_2n \approx \mathbf{O(n \log_2 n)}$$

Hoặc $O(n \log n)$

15

ĐỘ PHỨC TẠP QUICK SORT

TRƯỜNG HỢP	ĐỘ PHỨC TẠP
Tốt nhất	$O(n \log n)$
Xấu nhất	$O(n^2)$





TÀI LIỆU THAM KHẢO

- 1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest, Clifford Stein**, (Chapter 2, 3) *Introduction to Algorithms*, Third Edition, 2009.
- 2. Adam Drozdek**, (Chapter 9) *Data Structures and Algorithms in C++*, Fourth Edition, CENGAGE Learning, 2013.
- 3. Lê Xuân Trường**, (Chapter 2) *Cấu trúc dữ liệu*, NXB Trường Đại học Mở TP-HCM, 2016.

BÀI TẬP CHƯƠNG 3

Bài 1: Thực hiện mô tả từng bước quá trình sắp xếp thứ tự dãy số nguyên bằng thuật toán Quick Sort

Bài 2: Quản lý danh sách đặc 100 phần tử kiểu số nguyên (int)

-  2.1. Khai báo cấu trúc danh sách.
-  2.2. Viết thủ tục nhập danh sách.
-  2.3. Viết thủ tục xuất danh sách
-  2.4. Viết thủ tục sắp xếp danh sách theo thứ tự tăng dần bằng thuật toán Quick Sort

Bài tập thêm

Bài 3: Thực hiện đánh giá độ phức tạp của thuật toán QuickSort

Bài 4: Quản lý danh sách liên kết đơn (các phần tử kiểu số nguyên)

-  Viết thủ tục sắp xếp danh sách theo thứ tự tăng dần bằng thuật toán Quick Sort