

Cấu trúc dữ liệu và giải thuật

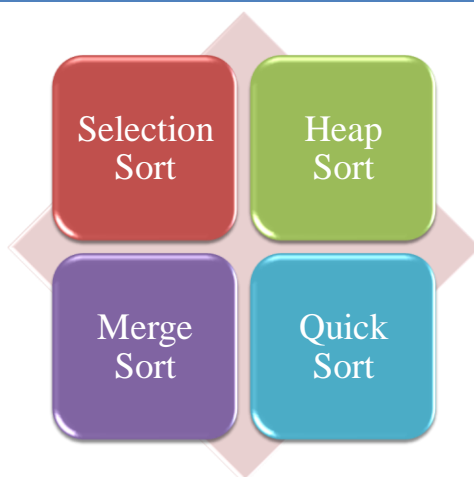
CÁC THUẬT TOÁN SẮP XẾP

Giảng viên:

Văn Chí Nam – Nguyễn Thị Hồng Nhung – Đặng Nguyễn Đức Tiến

Nội dung

2



Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Giới thiệu

Bài toán sắp xếp
Các thuật toán sắp xếp

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Giới thiệu

- Bài toán sắp xếp: Sắp xếp là quá trình xử lý một danh sách các phần tử để đặt chúng theo một thứ tự thỏa yêu cầu cho trước
- Ví dụ: danh sách trước khi sắp xếp:
 $\{1, 25, 6, 5, 2, 37, 40\}$
Danh sách sau khi sắp xếp:
 $\{1, 2, 5, 6, 25, 37, 40\}$
- Thông thường, sắp xếp giúp cho việc tìm kiếm được nhanh hơn.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Giới thiệu

5

◉ Các phương pháp sắp xếp thông dụng:

- ▣ Buble Sort
- ▣ **Selection Sort**
- ▣ Insertion Sort
- ▣ **Quick Sort**
- ▣ **Merge Sort**
- ▣ **Heap Sort**
- ▣ Radix Sort



Cần tìm hiểu các phương pháp sắp xếp và lựa chọn phương pháp phù hợp khi sử dụng.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

6

Sắp xếp chọn

Selection Sort

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Ý tưởng

7

- ◉ Mô phỏng cách sắp xếp tự nhiên nhất trong thực tế
 - ▣ Chọn phần tử nhỏ nhất và đưa về vị trí đúng là đầu dãy hiện hành.
 - ▣ Sau đó xem dãy hiện hành chỉ còn $n-1$ phần tử.
 - ▣ Lặp lại cho đến khi dãy hiện hành chỉ còn 1 phần tử.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán

8

Các bước của thuật toán:

- ◉ **Bước 1.** Khởi gán $i = 0$.
- ◉ **Bước 2.** Bước lặp:
 - ▣ 2.1. Tìm $a[\min]$ nhỏ nhất trong dãy từ $a[i]$ đến $a[n-1]$
 - ▣ 2.2. Hoán vị $a[\min]$ và $a[i]$
- ◉ **Bước 3.** So sánh i và n :
 - ▣ Nếu $i \leq n$ thì *tăng i thêm 1* và lặp lại bước 2.
 - ▣ Ngược lại: Dừng thuật toán.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Ví dụ

9



Đánh giá

10

◉ Đánh giá giải thuật:

▣ Số phép so sánh:

- Tại lượt i bao giờ cũng cần (n-i-1) số lần so sánh
- Không phụ thuộc vào tình trạng dãy số ban đầu

Số phép so sánh =
$$\sum_{i=0}^{n-1} (n-i-1) = \frac{n(n-1)}{2}$$

Đánh giá

11

◉ Số phép gán:

▣ Tốt nhất: $\sum_{i=0}^{n-1} 4 = 4n$

▣ Xấu nhất:

$$\sum_{i=0}^{n-1} (4 + n - i - 1) = \frac{n(n+7)}{2}$$

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

12

Sắp xếp vun đống

Heap Sort

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Ý tưởng

13

- ◉ Ý tưởng: khi tìm phần tử nhỏ nhất ở bước i , phương pháp Selection sort không tận dụng được các thông tin đã có nhờ vào các phép so sánh ở bước $i-1 \rightarrow$ cần khắc phục nhược điểm này.
- ◉ J. Williams đã đề xuất phương pháp sắp xếp Heapsort.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Heap

14

- ◉ Định nghĩa Heap:
 - ▣ Giả sử xét trường hợp sắp xếp tăng dần, Heap được định nghĩa là một dãy các phần tử a_1, a_{1+1}, \dots, a_r thỏa: với mọi i thuộc $[1, r]$ (chỉ số bắt đầu từ 0)
$$a_i \geq a_{2i+1}$$
$$a_i \geq a_{2i+2} \quad \{(a_i, a_{2i+1}), (a_i, a_{2i+2}) \text{ là các cặp phần tử liên đới}\}$$

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Các tính chất của Heap

15

- Nếu a_1, a_{i+1}, \dots, a_r là một heap thì phần tử a_1 (đầu heap) luôn là phần tử lớn nhất.
- Mọi dãy a_i, a_{i+1}, \dots, a_r với $2i + 1 > r$ là heap.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán

16

- Giai đoạn 1: Hiệu chỉnh dãy ban đầu thành heap (bắt đầu từ phần tử giữa của dãy)
- Giai đoạn 2: sắp xếp dựa trên heap.
 - ▣ Bước 1: đưa phần tử lớn nhất về vị trí đúng ở cuối dãy
 - ▣ Bước 2:
 - Loại bỏ phần tử lớn nhất ra khỏi heap: $r = r - 1$
 - Hiệu chỉnh lại phần còn lại của dãy.
 - ▣ Bước 3: So sánh r và l :
 - Nếu $r > l$ thì lặp lại bước 2.
 - Ngược lại, dừng thuật toán.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Heap Sort

17

◉ Mã giả (Tựa ngôn ngữ lập trình C):

```
void HeapSort(int a[], int n)
{
    TaoHeap(a, n-1);
    r = n-1;
    while(r > 0)
    {
        HoanVi(a[0], a[r]);
        r = r - 1;
        HieuChinh(a, 0, r);
    }
}
```

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Heap Sort

18

◉ Mã giả:

```
void TaoHeap(int a[], int r)
{
    int l = r/2;
    while(l > 0)
    {
        HieuChinh(a, l, r);
        l = l - 1;
    }
}
```

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Heap Sort

19

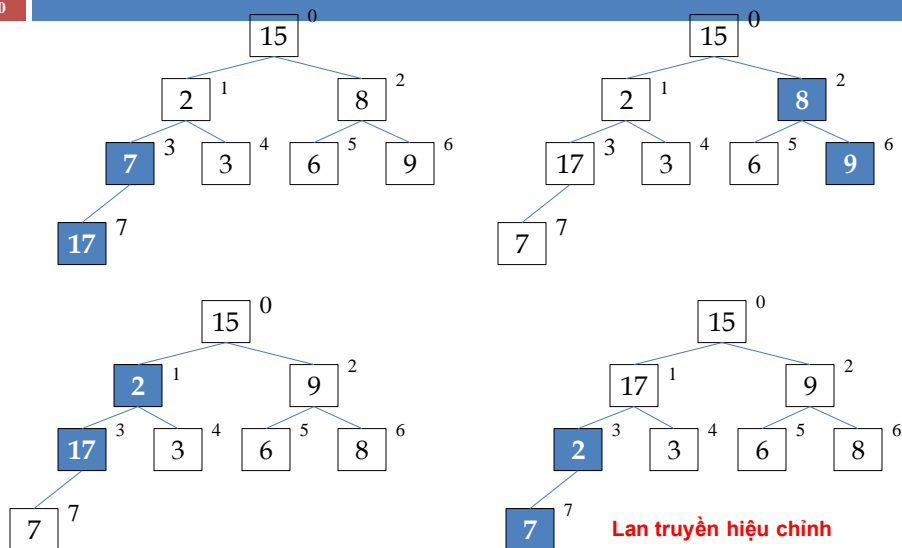
◉ Mã giả:

```
void HieuChinh(int a[], int l, int r)
{
    i = l; j = 2*i+1; x = a[i];
    while(j <= r)
    {
        if(có đủ 2 phần tử liên đới)
            //xác định phần tử liên đới lớn nhất
        if(a[j] < x) //thỏa quan hệ liên đới
            //dừng
        else
            //hiệu chỉnh
            //xét khả năng hiệu chỉnh lan truyền
    }
}
```

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

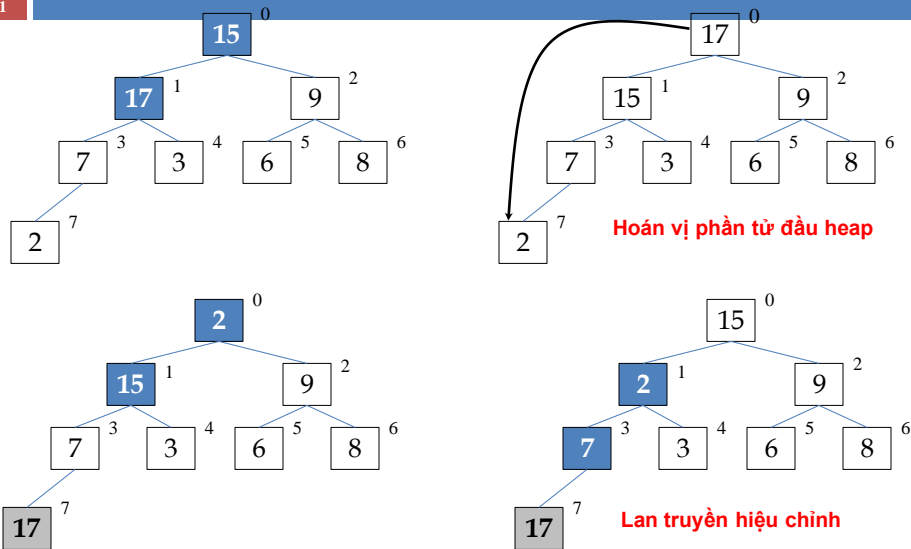
Ví dụ

20



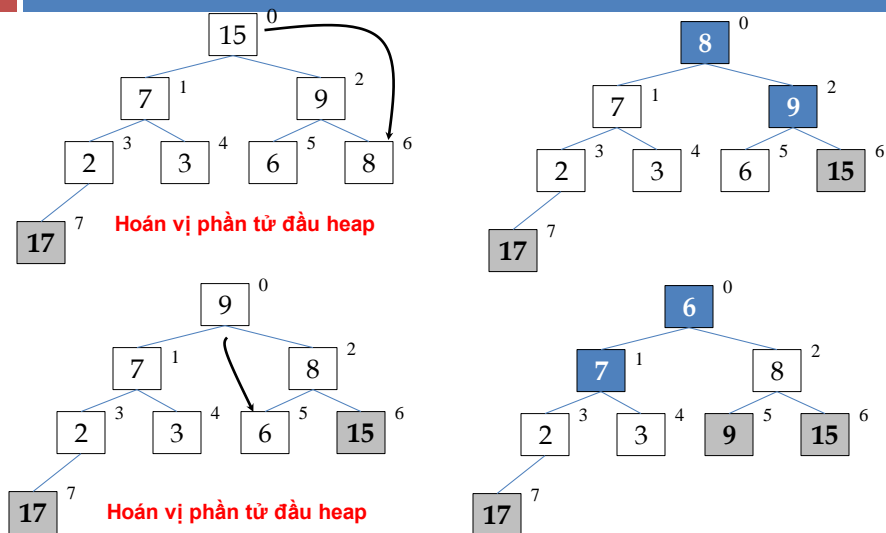
Ví dụ

21



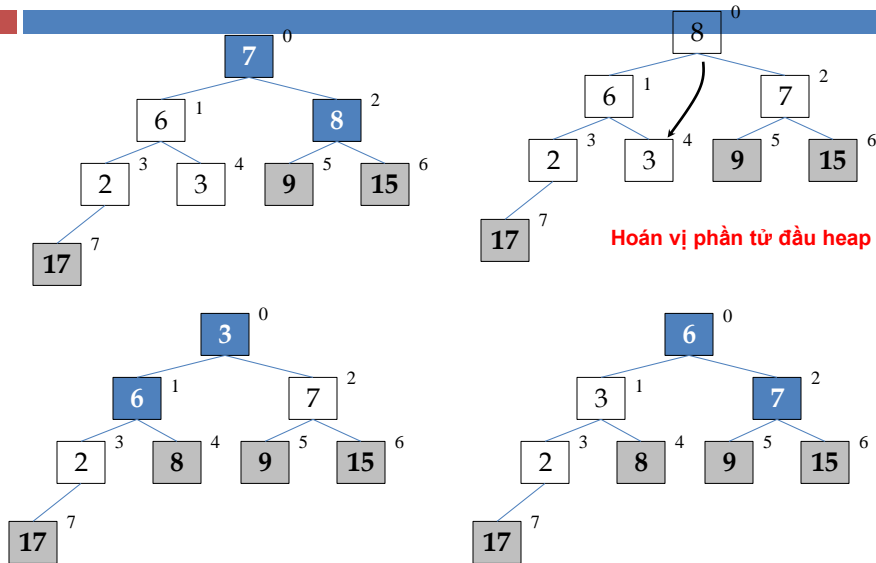
Ví dụ

22



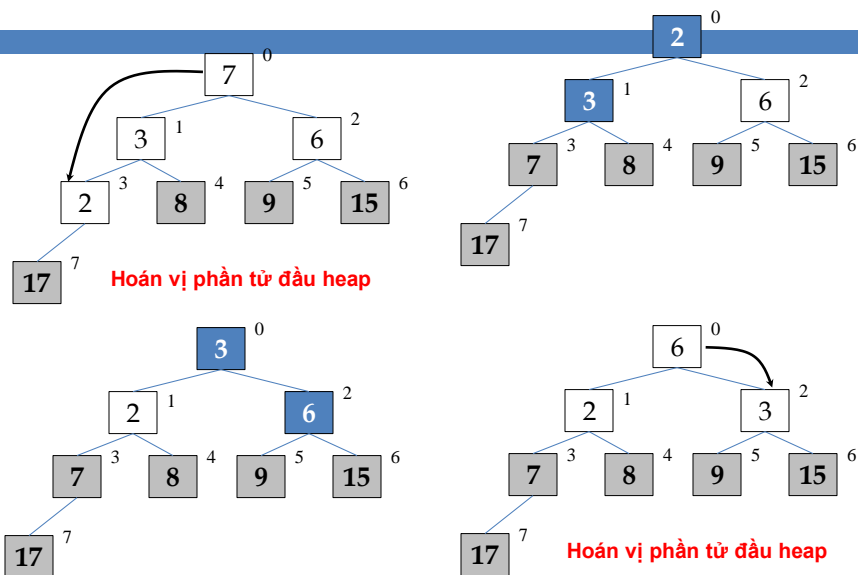
Ví dụ

23



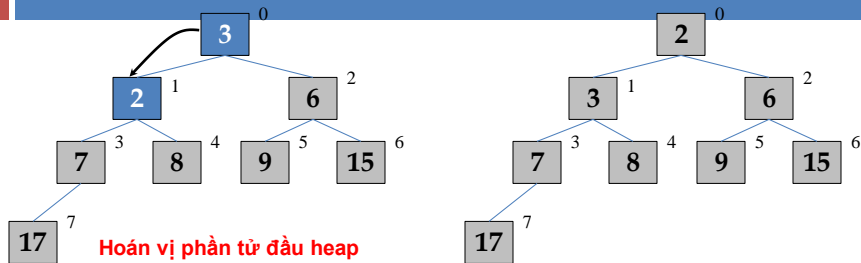
Ví dụ

24



Ví dụ

25



Mảng sau khi sắp xếp:



Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Heap Sort

26

- ◉ **Đánh giá giải thuật:**
 - ▣ Độ phức tạp của giải thuật trong trường hợp xấu nhất là $O(n \log_2 n)$

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Sắp xếp nhanh

Quick Sort

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Ý tưởng

- ◉ Giải thuật: dựa trên việc phân hoạch dãy ban đầu thành 2 phần:

- ▣ Dãy con 1: a_0, a_1, \dots, a_i có giá trị nhỏ hơn x
- ▣ Dãy con 2: a_j, \dots, a_{n-1} có giá trị lớn hơn x .

Dãy ban đầu được phân thành 3 phần:

$a_k < x$	$a_k = x$	$a_k > x$
$k = 0 \dots i$	$k = i+1 \dots j$	$k = j+1, \dots n-1$

- Phần 2 đã có thứ tự
- Phần 1, 3: cần sắp thứ tự, tiến hành phân hoạch từng dãy con theo cách phân hoạch dãy ban đầu

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Thuật toán – Giai đoạn phân hoạch

29

1. Chọn phần tử $a[k]$ trong dãy là giá trị mốc, $0 \leq k \leq r-1$
 - ▣ Gán $x = a[k]$, $i = l$, $j = r$.
 - ▣ Thường chọn phần tử ở giữa dãy: $k = (l+r)/2$
2. Phát hiện và hiệu chỉnh cặp phần tử $a[i]$, $a[j]$ sai vị trí:
 - ▣ 2.1. Trong khi $(a[i] < x)$, tăng i .
 - ▣ 2.2. Trong khi $(a[j] > x)$, giảm j .
 - ▣ 2.3. Nếu $i \leq j$ thì:
 - Hoán vị $a[i]$, $a[j]$,
 - Tăng i và giảm j
3. So sánh i và j :
 - ▣ Nếu $i < j$: lặp lại bước 2
 - ▣ Ngược lại: dừng phân hoạch.

Thuật toán Quick Sort

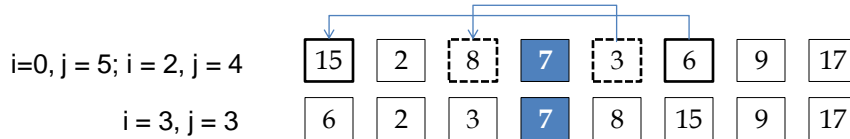
30

- ◉ Bước 1: phân hoạch dãy ban đầu thành 3 dãy:
 - ▣ Dãy 1: $a_1 \dots a_j < x$
 - ▣ Dãy 2: $a_{j+1} \dots a_{i-1} = x$
 - ▣ Dãy 3: $a_i \dots a_r > x$
- ◉ Bước 2: sắp xếp:
 - ▣ Nếu $l < j$: phân hoạch dãy $a_1 \dots a_j$
 - ▣ Nếu $i < r$: phân hoạch dãy $a_i \dots a_r$

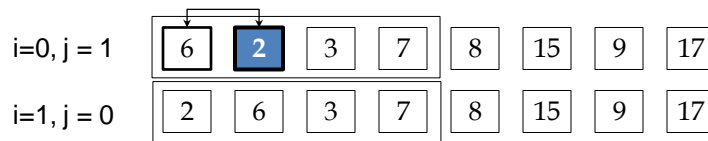
Ví dụ

31

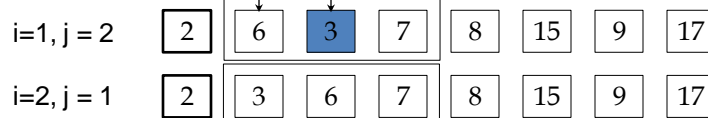
Phân hoạch dãy ban đầu: $l = 0, r = 7, x = a[3]$



Phân hoạch đoạn $l = 0, r = 3, x = a[1]$



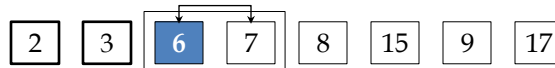
Phân hoạch đoạn $l = 1, r = 3, x = a[2]$



Ví dụ (tt)

32

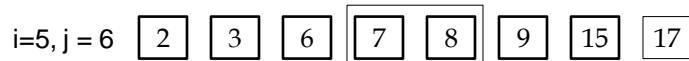
○ Phân hoạch đoạn $l = 2, r = 3, x = a[2]$



○ Phân hoạch đoạn $l = 3, r = 7, x = a[5]$



○ Phân hoạch đoạn $l = 3, r = 4, x = a[3]$



○ Phân hoạch đoạn $l = 5, r = 7, x = a[6]$



Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Bài tập

33

- ◉ Chạy tay thuật toán Quick Sort để sắp xếp mảng A trong 2 trường hợp tăng dần và giảm dần.

$A = \{2, 9, 5, 12, 20, 15, -8, 10\}$

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Quick Sort

34

- ◉ Đánh giá giải thuật:
 - ▣ Hiệu quả phụ thuộc vào việc chọn giá trị mốc
 - Tốt nhất là phần tử median.
 - Nếu phần tử mốc là cực đại hay cực tiểu thì việc phân hoạch không đồng đều.
 - ▣ Bảng tổng kết:

	Độ phức tạp
Tốt nhất	$n \cdot \log(n)$
Trung bình	$n \cdot \log(n)$
Xấu nhất	n^2

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

35

Sắp xếp trộn

Merge Sort

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Giới thiệu

36

- ◉ Thực hiện theo hướng chia để trị.
- ◉ Do John von Neumann đề xuất năm 1945.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Giải thuật

37

- ◉ Nếu dãy có chiều dài là 0 hoặc 1: đã được sắp xếp.
- ◉ Ngược lại:
 - ▣ Chia dãy thành 2 dãy con (chiều dài tương đương nhau).
 - ▣ Sắp xếp trên từng dãy con bằng thuật toán Merge Sort.
 - ▣ Trộn 2 dãy con (đã được sắp xếp) thành một dãy mới đã được sắp xếp.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

Giải thuật

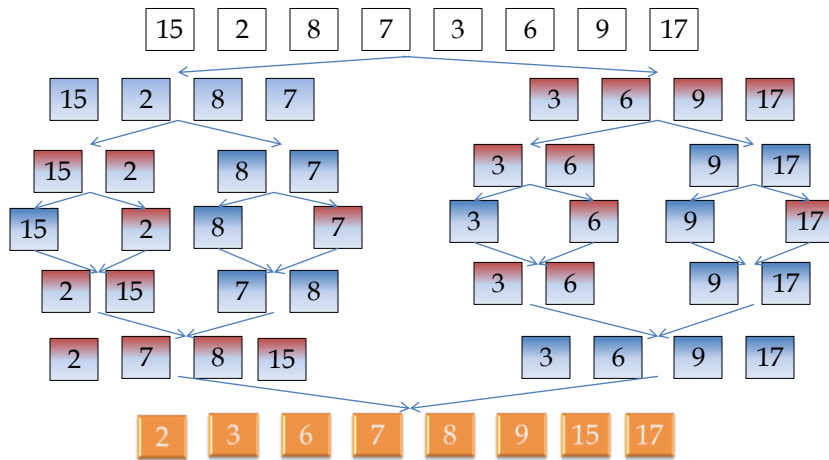
38

- ◉ Input: Dãy A và các chỉ số left, right (sắp xếp dãy A gồm các phần tử có chỉ số từ *left* đến *right*).
 - ◉ Output: Dãy A đã được sắp xếp
- ```
MergeSort(A, left, right)
{
 if (left < right) {
 mid = (left + right)/2;
 MergeSort(A, left, mid);
 MergeSort(A, mid+1, right);
 Merge(A, left, mid, right);
 }
}
```

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

## Ví dụ

39



Cấu trúc dữ liệu và giải thuật – HCMUS 2011

## Đánh giá

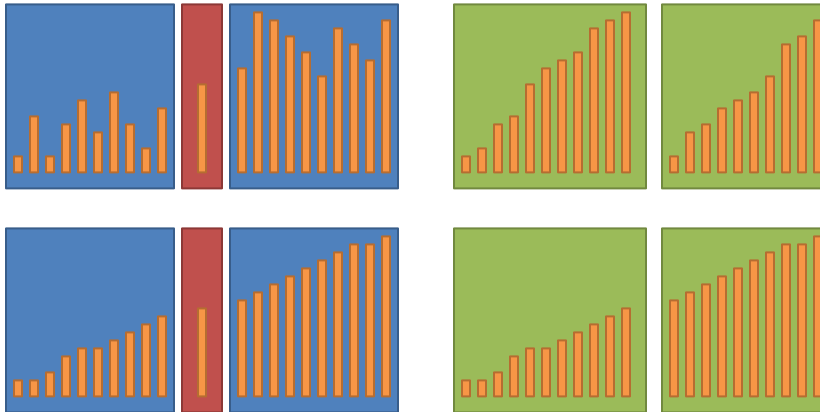
40

- Số lần chia các dãy con:  $\log_2 n$
- Chi phí thực hiện việc trộn hai dãy con đã sắp xếp tỷ lệ thuận với  $n$ .
- Chi phí của Merge Sort là  $O(n \log_2 n)$
- Thuật toán không sử dụng thông tin nào về đặc tính của dãy cần sắp xếp  $\Rightarrow$  chi phí thuật toán là không đổi trong mọi trường hợp

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

## So sánh tư tưởng sắp xếp giữa Quick sort và Merge sort

41



Cấu trúc dữ liệu và giải thuật – HCMUS 2011

42

## Kết luận

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

## Kết luận

43

- Các thuật toán Bubble sort, Selection sort, Insertion sort
  - ▣ Cài đặt thuật toán đơn giản.
  - ▣ Chi phí của thuật toán cao:  $O(n^2)$ .
- Heap sort được cải tiến từ Selection sort nhưng chi phí thuật toán thấp hơn hẳn ( $O(n\log_2 n)$ )

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

## Kết luận

44

- Các thuật toán Quick sort, Merge sort là những thuật toán theo chiến lược chia để trị.
  - ▣ Cài đặt thuật toán phức tạp
  - ▣ Chi phí thuật toán thấp:  $O(n\log_2 n)$
  - ▣ Rất hiệu quả khi dùng danh sách liên kết.
  - ▣ Trong thực tế, Quick sort chạy nhanh hơn hẳn Merge sort và Heap sort.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

## Kết luận

45

- ◉ Người ta chứng minh  $O(n \log_2 n)$  là ngưỡng chặn dưới của các thuật toán sắp xếp dựa trên việc so sánh giá trị của các phần tử.

Cấu trúc dữ liệu và giải thuật – HCMUS 2011

46

## Hỏi và Đáp

Cấu trúc dữ liệu và giải thuật – HCMUS 2011