CHƯƠNG 3. CÁC THUẬT TOÁN TÌM KIẾM & SẮP XẾP

SẮP XẾP TRỘN (MERGE SORT)

ThS. Nguyễn Chí Hiếu

2021

NỘI DUNG

NỘI DUNG

Thuật toán sắp xếp trộn (MergeSort)

Ý tưởng

- Trong giải thuật sắp xếp trộn *(MergeSort)*, mỗi dãy $a_0, a_2, \cdots, a_{n-1}$ bất kỳ đều có thể coi như là một tập hợp các dãy con liên tiếp mà mỗi dãy con đều đã có thứ tự.
- Các dãy con này trộn với nhau sẽ tạo thành dãy có thứ tự.

Ví dụ 1

Cho dãy số $12, 2, 8, 5, 1, 6, 4, 15$ gồm 5 dãy con có thứ tự:

$$\{12\}, \{2, 8\}, \{5\}, \{1, 6\}, \{4, 15\}$$

Trộn các dãy con tạo thành dãy có thứ tự.

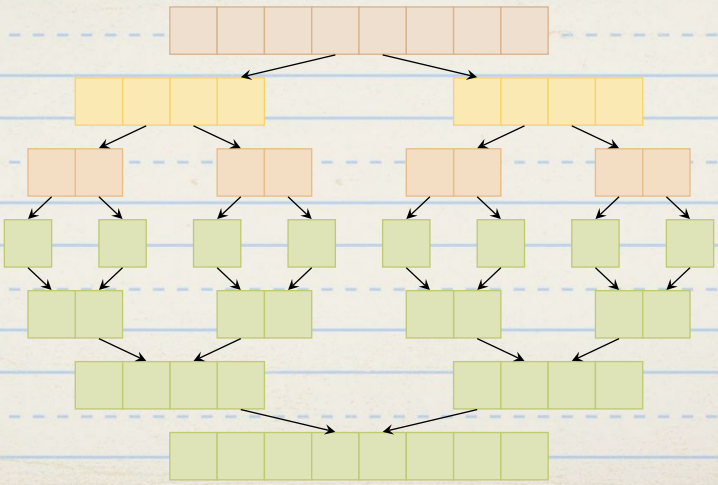
$$1, 2, 4, 5, 8, 6, 12, 15$$

Thuật toán sắp xếp trộn (MergeSort)

Áp dụng phương pháp chia để trị

- Chia: chia dãy a gồm n phần tử thành hai dãy con có kích thước khoảng $\frac{n}{2}$.
- Trị: nếu dãy con chứa từ hai phần tử trở lên thì gọi đệ quy giải thuật MergeSort đối với dãy con đang xét.
- Tổng hợp: trộn hai dãy con với nhau thành một dãy có thứ tự.

Thuật toán sắp xếp trộn (MergeSort)



Thuật toán sắp xếp trộn (MergeSort)

Thuật toán 1: MergeSort(a[], left, right)

- Đầu vào: mảng a, left và right tương ứng vị trí bắt đầu và kết thúc của mảng đang xét.
- Đầu ra: mảng a có thứ tự tăng dần.

```
1   if left < right
2     middle ← (left + right) / 2
3     MergeSort(a, left, middle)
4     MergeSort(a, middle + 1, right)
5     Merge(a, left, middle, right)
```

Giải thích

- Dòng 1: nếu mảng nhiều hơn một phần tử thì chia thành 2 mảng con bên trái và bên phải tại phần tử giữa.
- Dòng 3, 4: gọi đệ quy giải thuật với mảng con bên trái và bên phải.
- Dòng 5: kết hợp 2 mảng con thành mảng có thứ tự.

Thuật toán sắp xếp trộn (MergeSort)

Thuật toán 2: Merge(a[], left, middle, right)

- Đầu vào: mảng a, vị trí left, middle, right.
- Đầu ra: mảng a sau khi được trộn.

```
1   array b[left..right]
2   k ← left, i ← left, j ← middle + 1
3   while i ≤ middle or j ≤ right
4     if a[i] ≤ a[j]
5       b[k++] ← a[i++]
6     else
7       b[k++] ← a[j++]
8   while i ≤ middle
9     b[k++] ← a[i++]
10  while j ≤ right
11    b[k++] ← a[j++]
12  for k ← left to right
13    a[k] ← b[k]
```

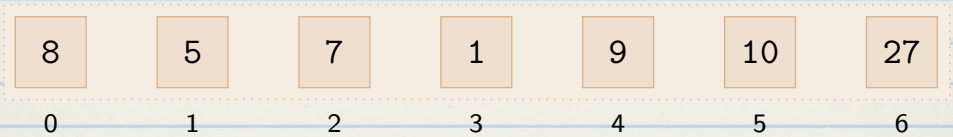
Thuật toán sắp xếp trộn (MergeSort)

Giải thích

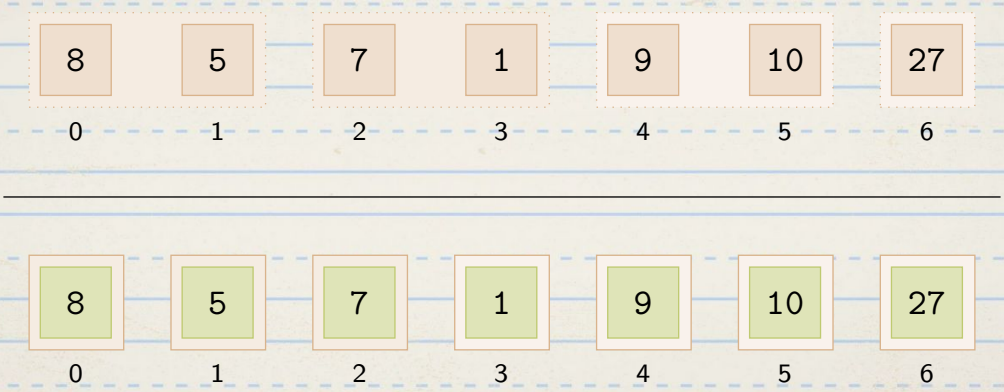
- Dòng 1: khai báo b là mảng tạm.
- Dòng 3 $\rightarrow$ 7: trường hợp hai mảng con bên trái và bên phải đều khác rỗng.
  - Dòng 4, 5: chép mảng bên trái vào b.
  - Dòng 6, 7: chép mảng bên phải vào b.
- Dòng 8, 9: chép phần còn lại của mảng bên trái vào b.
- Dòng 10, 11: chép phần còn lại của mảng bên phải vào b.
- Dòng 12, 13: chép mảng b vào mảng a. Kết thúc thuật toán.

Ví dụ

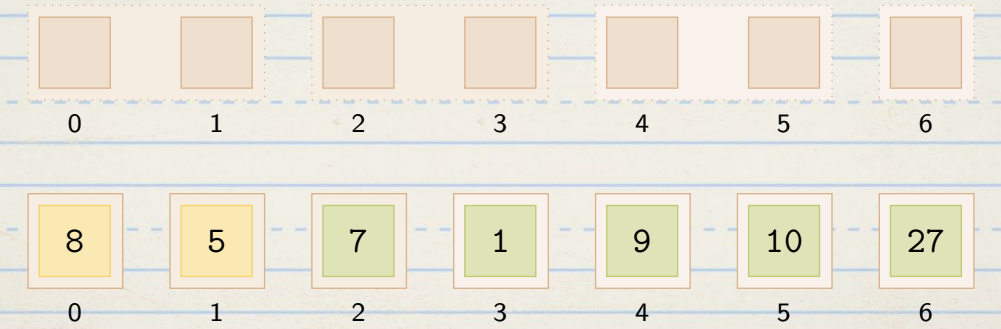
Cho dãy số a gồm 7 phần tử: $8, 5, 7, 1, 9, 10, 27$. Áp dụng giải thuật MergeSort sắp dãy a theo thứ tự tăng dần.



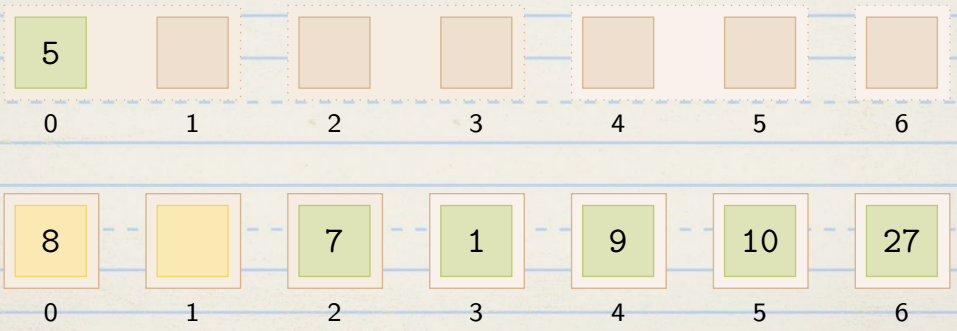
Ví dụ minh họa



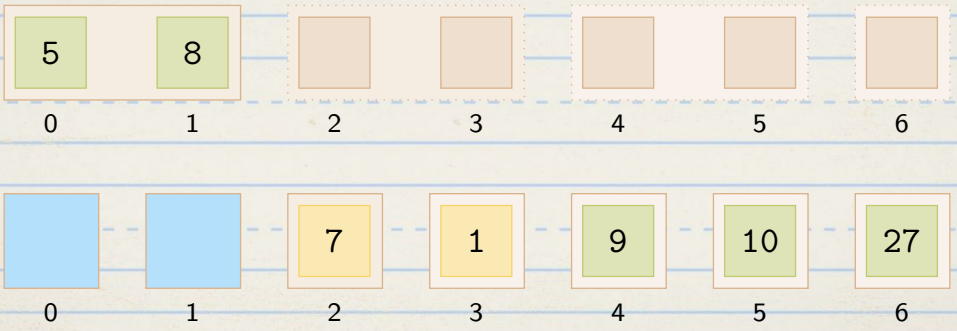
Ví dụ minh họa



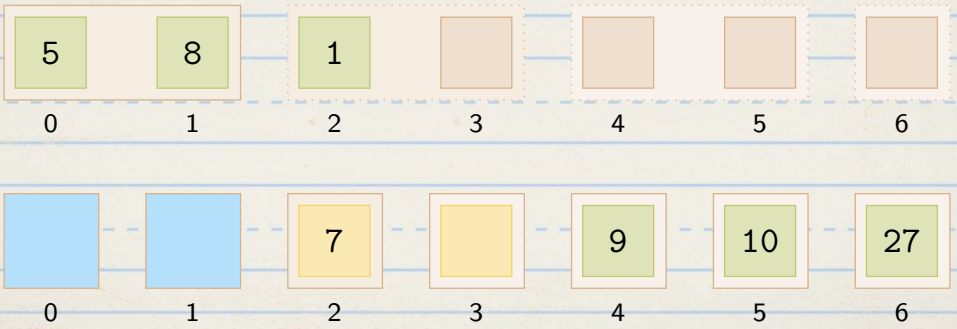
Ví dụ minh họa



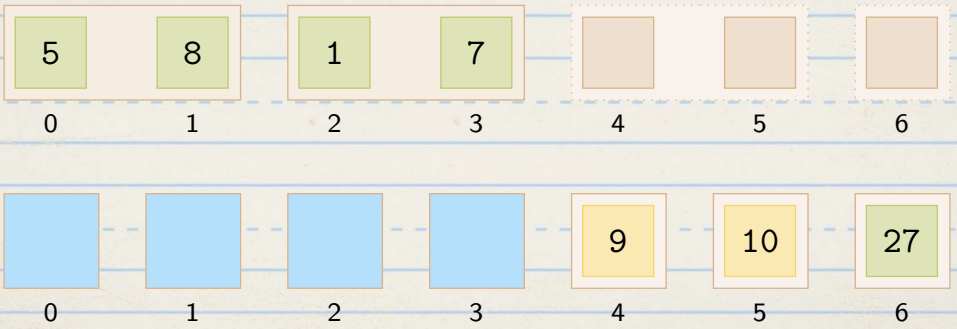
Ví dụ minh họa



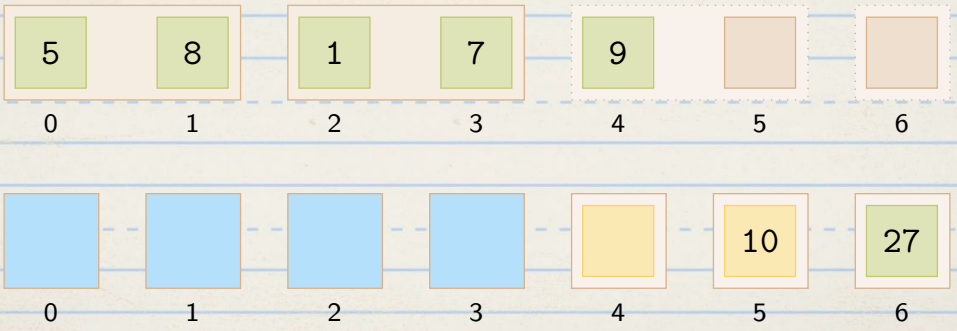
Ví dụ minh họa



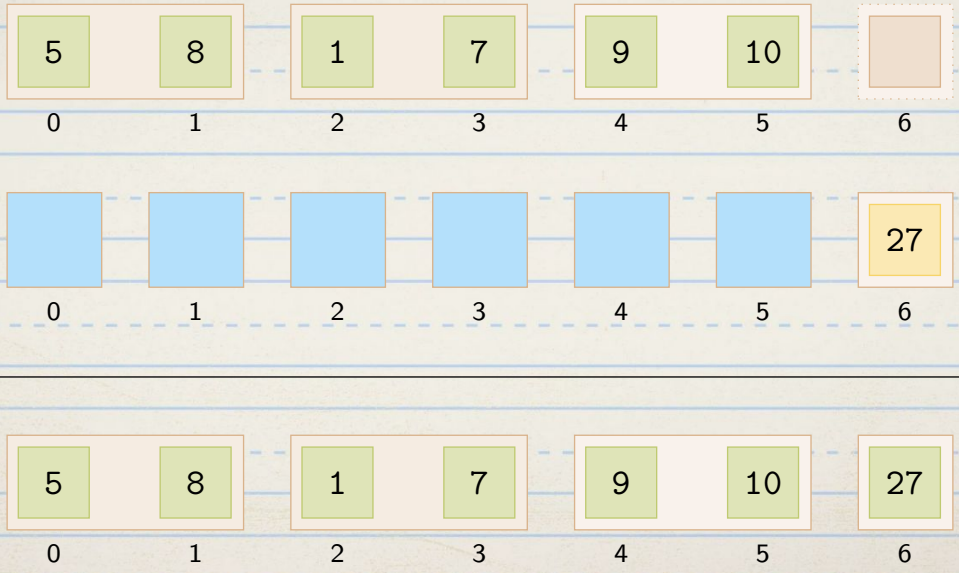
Ví dụ minh họa



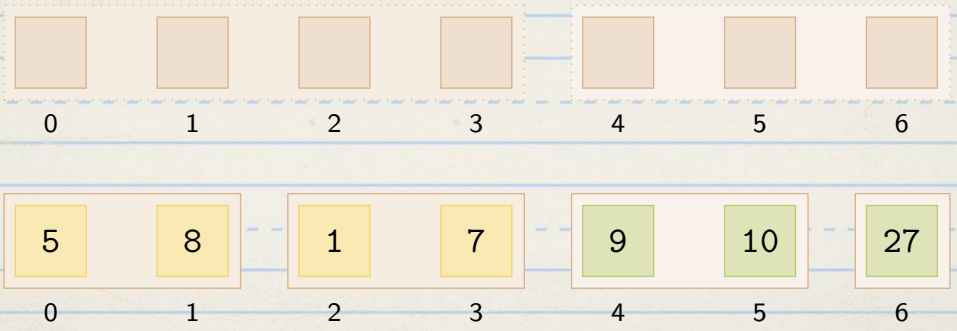
Ví dụ minh họa



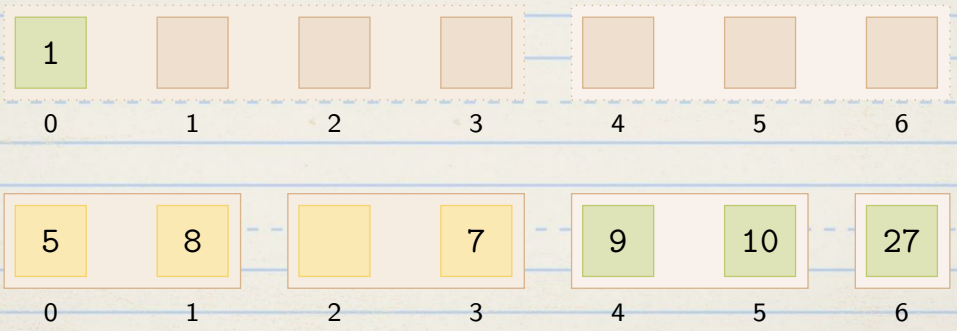
Ví dụ minh họa



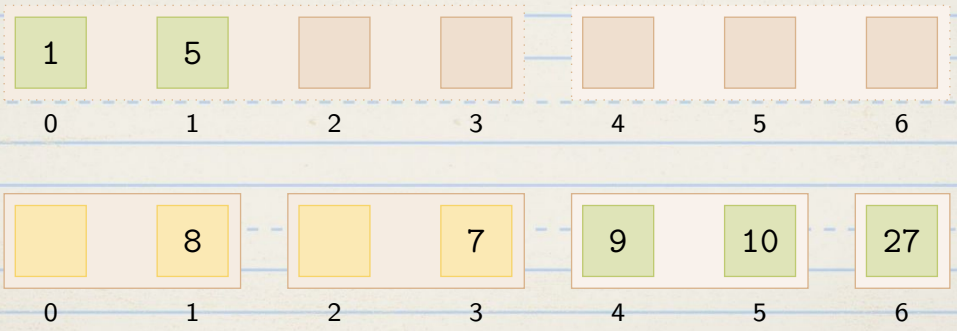
Ví dụ minh họa



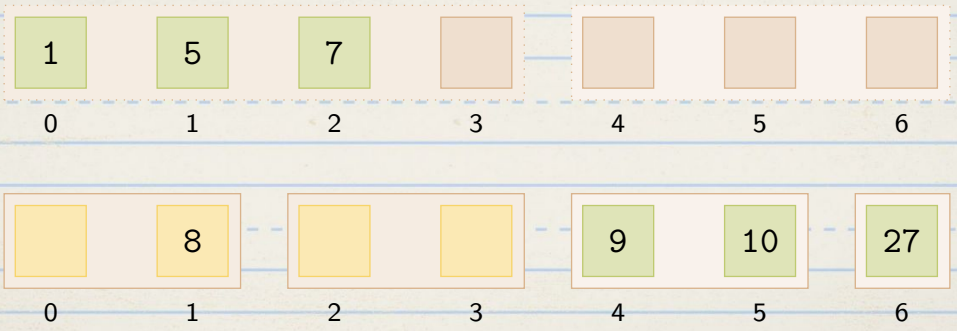
Ví dụ minh họa



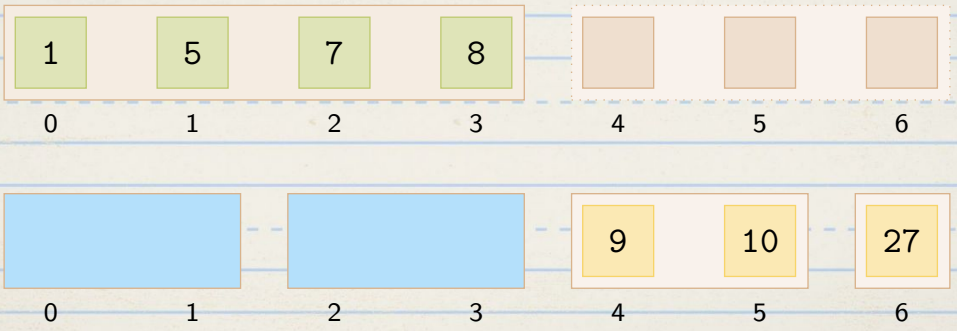
Ví dụ minh họa



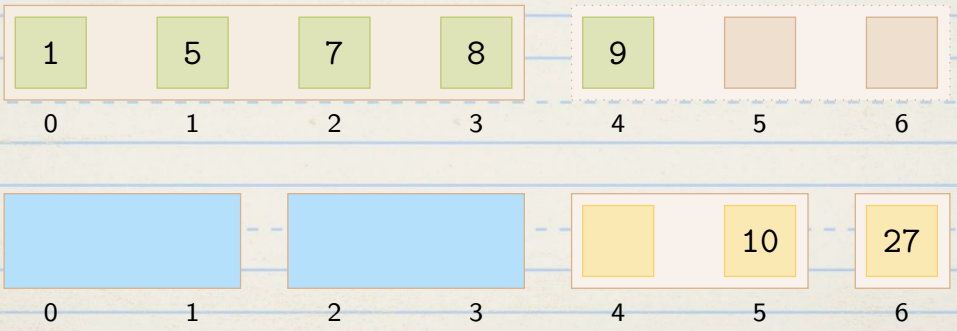
Ví dụ minh họa



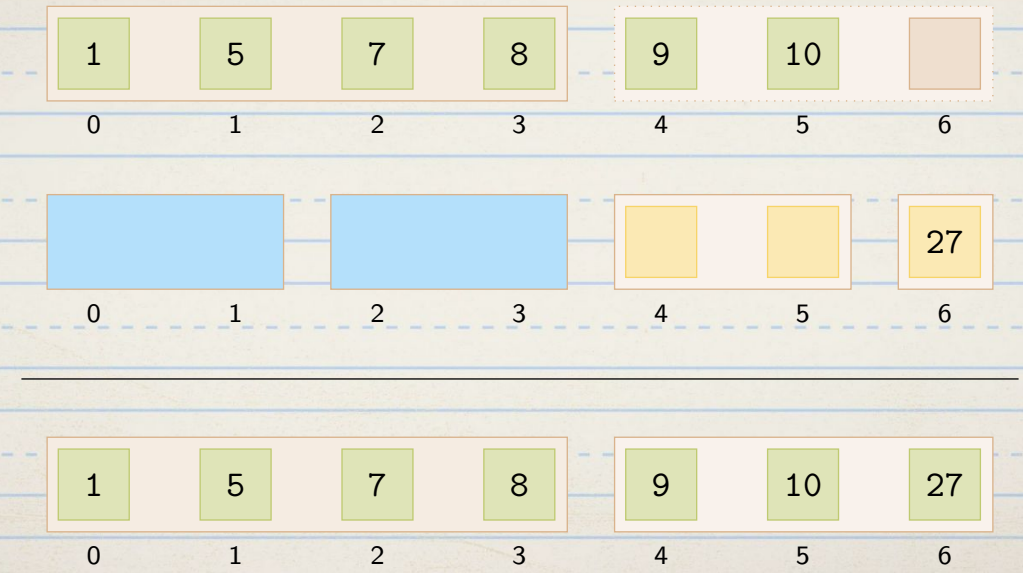
Ví dụ minh họa



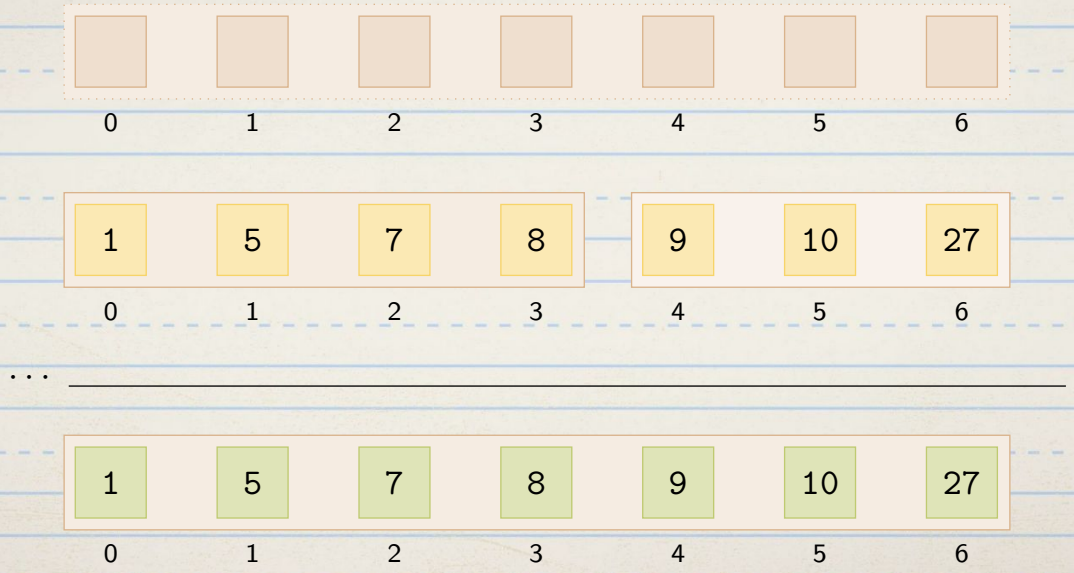
Ví dụ minh họa



Ví dụ minh họa



Ví dụ minh họa



NỘI DUNG

Đánh giá độ phức tạp

Trường hợp xấu nhất/tốt nhất

- Hàm MergeSort: nếu dãy có hơn một phần tử thì mỗi lần thực hiện sẽ chia thành 2 dãy con có $\frac{n}{2}$ phần tử. Thời gian thực hiện của dãy con là $T\left(\frac{n}{2}\right)$.
- Hàm Merge: thời gian thực hiện là $O(n)$.

Đánh giá độ phức tạp

Trường hợp xấu nhất/tốt nhất

- Thời gian thực hiện thuật toán

$$T(n) = \begin{cases} 1 & ,n = 1 \\ T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + n & ,n > 1 \end{cases}$$

- Nếu n là lũy thừa của 2 (trộn 2 dãy/2-way) thì

$$T(n) = \begin{cases} 1 & ,n = 1 \\ 2T\left(\frac{n}{2}\right) + n & ,n > 1 \end{cases}$$

Đánh giá độ phức tạp

Trường hợp xấu nhất/tốt nhất

Chứng minh

$$\begin{aligned} T(n) &= 2T\left(\tfrac{n}{2}\right) + n \\ &= 4T\left(\tfrac{n}{4}\right) + n + n \\ &= 8T\left(\tfrac{n}{8}\right) + n + n + n \\ &\dots \\ &= nT\left(\frac{n}{2^k}\right) + k \cdot n \qquad ,n \geq 2^k \end{aligned}$$

Đánh giá độ phức tạp

Trường hợp xấu nhất/tốt nhất

Chứng minh

- Giả sử $n = 2^k$, thuật toán dừng đệ quy.

$$\begin{aligned} T(n) &= 2^k \cdot T(1) + k \cdot 2^k \\ &= 2^k + k \cdot 2^k \\ &= n + n \log n \end{aligned}$$

- Do đó,

$$T(n) = O(n \log n).$$

Bài tập

1. Áp dụng giải thuật MergeSort sắp dãy sau theo thứ tự tăng dần

$$8, 3, 2, 9, 7, 1, 5, 4$$

2. Xây dựng giải thuật MergeSort trộn 3-dãy *(3-way)*. Áp dụng giải thuật sắp dãy sau theo thứ tự tăng dần

$$2, 1, 5, 7, 4, 6, 9, 3, 8$$

Tài liệu tham khảo


Donald E. Knuth.
The Art of Computer Programming, Volume 3.
Addison-Wesley, 1998.


Dương Anh Đức, Trần Hạnh Nhi.
Nhập môn Cấu trúc dữ liệu và Thuật toán.
Đại học Khoa học tự nhiên TP Hồ Chí Minh, 2003.


Niklaus Wirth.
Algorithms + Data Structures = Programs.
Prentice-Hall, 1976.


Robert Sedgewick.
Algorithms in C.
Addison-Wesley, 1990.