

# HUST

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.





ĐẠI HỌC  
BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY  
OF SCIENCE AND TECHNOLOGY

# CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

TUẦN 1 : CÁC KHÁI NIỆM CƠ BẢN  
SoICT - HUST

ONE LOVE. ONE FUTURE.

# MỤC TIÊU

*Sau bài học này, người học có thể:*

1. Hiểu được một số khái niệm cơ bản về thuật toán
2. Biết ký hiệu tiệm cận dùng để đánh giá độ phức tạp thuật toán
3. Biết cách phân tích độ phức tạp của thuật toán

# MỤC LỤC

- 1 Ví dụ minh họa
- 2 Một số khái niệm cơ bản về thuật toán
- 3 Ký hiệu tiệm cận
- 4 Kỹ thuật phân tích thuật toán

# 1. Ví dụ minh họa

## ■ Bài toán tìm dãy con lớn nhất:

- Cho dãy số gồm  $n$  số:  $a_0, a_1, a_2, \dots, a_{n-1}$
- Dãy gồm liên tiếp các số  $a_i, a_{i+1}, \dots, a_j$  với  $0 \leq i \leq j \leq n - 1$  được gọi là **dãy con** của dãy đã cho và  $\sum_{k=i}^j a_k$  được gọi là **trọng lượng** của dãy con này
- **Hãy tìm trọng lượng lớn nhất của dãy con, tức là tìm cực đại giá trị  $\sum_{k=i}^j a_k$ .** Ta gọi dãy con có trọng lượng lớn nhất là **dãy con lớn nhất**.

# 1. Ví dụ minh họa

## ■ Bài toán tìm dãy con lớn nhất:

- Cho dãy số gồm  $n$  số:  $a_0, a_1, a_2, \dots, a_{n-1}$
- Dãy gồm liên tiếp các số  $a_i, a_{i+1}, \dots, a_j$  với  $0 \leq i \leq j \leq n - 1$  được gọi là **dãy con** của dãy đã cho và  $\sum_{k=i}^j a_k$  được gọi là **trọng lượng** của dãy con này
- **Hãy tìm trọng lượng lớn nhất của dãy con, tức là tìm cực đại giá trị  $\sum_{k=i}^j a_k$ .** Ta gọi dãy con có trọng lượng lớn nhất là **dãy con lớn nhất**.

## ■ **Ví dụ:** Cho dãy số $-2, 11, -4, 13, -5, 2$ thì cần đưa ra câu trả lời là $20$ (dãy con lớn nhất là $11, -4, 13$ với giá trị $= 11 + (-4) + 13 = 20$ )

# 1. Ví dụ minh họa

## ■ Cách 1: Duyệt toàn bộ

- Duyệt tất cả các dãy con có thể có của dãy đã cho:  $a_i, a_{i+1}, \dots, a_j$  với  $0 \leq i \leq j \leq n - 1$ , và tính tổng của mỗi dãy con để tìm ra trọng lượng lớn nhất.

```
int maxSum = a[0];
for (int i = 0; i <= n-1; i++) {
    for (int j = i; j <= n-1; j++) {
        int sum = 0;
        for (int k = i; k <= j; k++)
            sum += a[k];
        if (sum > maxSum) maxSum = sum;
    }
}
```

# 1. Ví dụ minh họa

## Cách 1: Duyệt toàn bộ

Duyệt tất cả các dãy con có thể có của dãy đã cho:  $a_i, a_{i+1}, \dots, a_j$  với  $0 \leq i \leq j \leq n - 1$ , và tính tổng của mỗi dãy con để tìm ra trọng lượng lớn nhất.

**Phân tích thuật toán:** Ta sẽ tính số lượng phép cộng mà thuật toán phải thực hiện, tức là đếm xem dòng lệnh `sum += a[k]` phải thực hiện bao nhiêu lần. Số lượng phép cộng là:

```
int maxSum = a[0];
for (int i = 0; i<=n-1; i++) {
    for (int j = i; j<=n-1; j++) {
        int sum = 0;
        for (int k=i; k<=j; k++) sum += a[k];
        if (sum > maxSum) maxSum = sum;
    }
}
```

$$\begin{aligned} \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} (j - i + 1) &= \sum_{i=0}^{n-1} (1 + 2 + \dots + (n - i)) = \sum_{i=0}^{n-1} \frac{(n - i)(n - i + 1)}{2} \\ &= \frac{1}{2} \sum_{k=1}^n k(k + 1) = \frac{1}{2} \left[ \sum_{k=1}^n k^2 + \sum_{k=1}^n k \right] = \frac{1}{2} \left[ \frac{n(n + 1)(2n + 1)}{6} + \frac{n(n + 1)}{2} \right] \\ &= \frac{n^3}{6} + \frac{n^2}{2} + \frac{n}{3} \end{aligned}$$

# 1. Ví dụ minh họa

## ■ Cách 2: Duyệt toàn bộ có cải tiến:

Index i	0	1	2	3	4	5
a[i]	-2	11	-4	13	-5	2

$i = 0$ :

$$\begin{aligned} & 9 + (-4) = 5 \quad 5 \quad 18 + (-5) = 13 \quad 13 \\ & (-2), (-2, 11), \underbrace{(-2, 11, -4)}, \underbrace{(-2, 11, -4, 13)}, \underbrace{(-2, 11, -4, 13, -5)}, \underbrace{(-2, 11, -4, 13, -5, 2)} \\ & -2 + 11 = 9 \quad 9 \quad 5 + 13 = 18 \quad 18 \end{aligned}$$

- Nhận thấy, ta có thể tính tổng các phần tử từ vị trí  $i$  đến  $j$  từ tổng của các phần tử từ  $i$  đến  $j-1$  chỉ bằng 1 phép cộng:

$$\sum_{k=i}^j a[k] = a[j] + \sum_{k=i}^{j-1} a[k]$$

Tổng các phần tử từ  $i$  đến  $j$

Tổng các phần tử từ  $i$  đến  $j-1$

# 1. Ví dụ minh họa

$$\sum_{k=i}^j a[k] = a[j] + \sum_{k=i}^{j-1} a[k]$$



Tổng các phần tử từ i đến j



Tổng các phần tử từ i đến j-1

```
int maxSum = a[0];
for (int i=0; i<=n-1; i++) {
    for (int j=i; j<=n-1; j++) {
        int sum = 0;
        for (int k=i; k<=j; k++) sum += a[k];
        if (sum > maxSum) maxSum = sum;
    }
}
```



```
int maxSum = a[0];
for (int i=0; i<=n-1; i++) {
    int sum = 0;
    for (int j=i; j<=n-1; j++) {
        sum += a[j];
        if (sum > maxSum) maxSum = sum;
    }
}
```

# MỤC LỤC

- 1 Ví dụ minh họa
- 2 Một số khái niệm cơ bản về thuật toán
- 3 Ký hiệu tiệm cận
- 4 Kỹ thuật phân tích thuật toán

## 2. Một số khái niệm cơ bản về thuật toán

Nội dung slide

Nội dung 1

Nội dung 2

Nội dung 3

# MỤC LỤC

- 1 Ví dụ minh họa
- 2 Một số khái niệm cơ bản về thuật toán
- 3 Ký hiệu tiệm cận
- 4 Kỹ thuật phân tích thuật toán

### 3. Ký hiệu tiệm cận

Nội dung slide

Nội dung 1

Nội dung 2

Nội dung 3

# MỤC LỤC

- 1 Ví dụ minh họa
- 2 Một số khái niệm cơ bản về thuật toán
- 3 Ký hiệu tiệm cận
- 4 Kỹ thuật phân tích thuật toán

## 4. Kỹ thuật phân tích thuật toán

Nội dung slide

Nội dung 1

Nội dung 2

Nội dung 3



A large, stylized graphic of the letters "HUST" is composed of numerous small red dots arranged in a circular, swooping pattern against a dark blue background.

**HUST**

**THANK YOU!**



**HUST**

**CẢM ƠN  
ĐÃ LẮNG NGHE!**