

# Phương pháp thiết kế thuật toán: Divide and Conquer

## **Thành viên nhóm 2:**

- + Nguyễn Xuân Minh – 19521848
- + Lê Văn Phước – 19522054
- + Nguyễn Đức Thắng – 19522206
- + Trần Hữu Nguyên - 19521921

## **Giáo viên hướng dẫn:**

**Ths. Nguyễn Thanh Sơn**

# M U C L U C

1.

**Đặc điểm bài toán & dạng thuật toán phổ quát.**

2.

**Ưu & nhược điểm của phương pháp Divide and Conquer.**

3.

**Các bài toán minh họa.**

# 1

## Đặc điểm của bài toán

- ▶ Bài toán ban đầu có thể chia thành các bài toán con nhỏ hơn.
- ▶ Kết hợp kết quả của các bài toán nhỏ hơn có thể tìm ra giải pháp cho bài toán ban đầu.

# 1

## Đặc điểm của bài toán

- ▶ **Các bài toán chia để trị thường được phân chia và hoạt động theo nguyên tắc sau:**
- ▶ 1. Một bài toán được chia thành nhiều bài toán con cùng loại, lý tưởng nhất là kích thước bài toán con bằng nhau.
- ▶ 2. Các bài toán con được giải quyết (thường là đệ quy, mặc dù đôi khi một thuật toán khác được sử dụng, đặc biệt khi bài toán con trở nên đủ nhỏ).
- ▶ 3. Nếu cần thiết, các giải pháp cho các bài toán con được kết hợp để có được một giải pháp đến vấn đề ban đầu.

# 1

## Dạng bài toán phổ quát.

### Gồm 3 bước chính:

- **Chia (Divide):** chia bài toán ban đầu thành các bài toán con cho đến khi không thể chia được nữa.
- **Giải quyết bài toán con(Conquer):** Xử lý các bài toán con ở trên.
- **Kết hợp (Combine):** kết hợp kết quả bài toán con để đưa ra đáp án bài toán ban đầu.

# 2

## Ưu & nhược điểm của phương pháp

### Ưu điểm

- Giúp các vấn đề nhỏ được xử lý gần như một lúc, giúp giảm thiểu thời gian và chi phí thực thi gấp nhiều lần.
- Tận dụng bộ nhớ đệm (cache) một cách hiệu quả. Có thể giải quyết được trên bộ nhớ cache, không cần gửi thông tin đến bộ nhớ truy cập.

# 2

## Ưu & nhược điểm của phương pháp

### Nhược điểm

- Chia để trị không thể lưu lại kết quả của những vấn đề đã giải quyết cho lần yêu cầu tiếp theo.
- Khó khăn trong việc chia bài toán lớn thành các **bài toán nhỏ** và các bài toán nhỏ phải có **cùng cách giải** nếu không sự phức tạp sẽ tăng lên gấp bội.

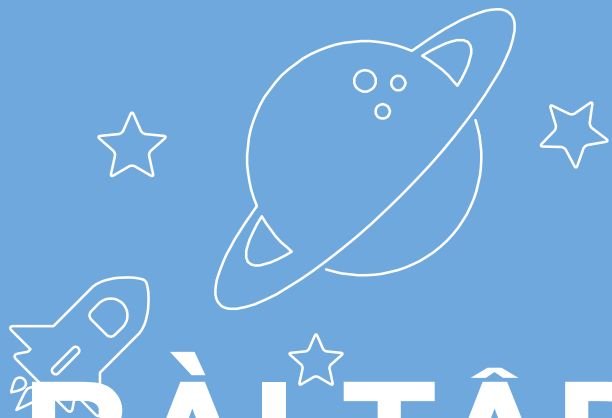
# 3

Một số bài  
toán áp  
dụng chia  
để trị

- ▶ Mergesort
- ▶ Quicksort
- ▶ Binary Tree Traversals and Related Properties
- ▶ Multiplication of Large Integers and Strassen's Matrix Multiplication.
- ▶ The closest-Pair and Convex-Hull Problems by Divide-and-Conquer



4



# BÀI TẬP ÁP DỤNG

# 4

A close-up photograph of a hand holding a blue pen, poised to write on a white sheet of paper. The hand is wearing a grey, textured sweater sleeve. The background is softly blurred, showing more of the paper and the pen.

## Bài 1: Tính lũy thừa

Tính  $a^n$ , với  $n \in \mathbb{N}$

# 4



## Bài 1: Tính lũy thừa

### Cách đơn giản, dễ thực hiện

```
def luythua(a,n):
```

```
    x = a
```

```
    for _ in
```

```
        range(1,n):
```

```
            x= x*a
```

```
    return x
```

$$T(n) = n$$

$$\Rightarrow T(n) \in$$

$\theta(n)$  tuyến tính

# 4

## Bài 1: Tính lũy thừa

### Áp dụng giải thuật Divide & Conquer

Thuật toán áp dụng chia để trị:

$$a^n = \begin{cases} a^{n/2} \times a^{n/2} & n \text{ chẵn} \\ a^{(n-1)/2} \times a^{(n-1)/2} \times a & n \text{ lẻ} \end{cases}$$

$$T(n) = 2 * T\left(\frac{n}{2}\right)$$

$$\Rightarrow T(n) \in \theta(n)$$

**PowerN**(a, n)

1    **if** n = 0 **return** 1

2    **if** n % 2 = 0

3        **return** **PowerN**(a,  $\frac{n}{2}$ ) × **PowerN**(a,  $\frac{n}{2}$ )

4    **else**

5        **return** **PowerN**(a,  $\frac{n-1}{2}$ ) × **PowerN**(a,  $\frac{n-1}{2}$ ) × a

***Đây là cách giải sai lầm, do  
không tối ưu được số lần gọi  
đệ quy nên độ phức tạp vẫn  
là  $O(n)$ .***

# 4

## Bài 1: Tính lũy thừa

### Áp dụng giải thuật Divide & Conquer

**PowerN**( $a, n$ )

```
1  if  $n = 0$  return 1
2  if  $n \% 2 = 0$ 
3       $p \leftarrow \text{PowerN}\left(a, \frac{n}{2}\right)$ 
4      return  $p \times p$ 
5  else
6       $p \leftarrow \text{PowerN}\left(a, \frac{n-1}{2}\right)$ 
7      return  $p \times p \times a$ 
```

$$T(n) = T\left(\frac{n}{2}\right)$$
$$\Rightarrow T(n) \in \theta(\log n)$$

# 4

A close-up photograph of a hand holding a blue pen, poised to write on a white sheet of paper. The hand is wearing a grey, textured sweater sleeve. The background is slightly blurred, showing more of the paper and the pen.

## Bài 2: Dãy Fibonacci

$$F_n = \begin{cases} n & n = 0, 1 \\ F_{n-1} + F_{n-2} & n \geq 2 \end{cases}$$

# 4

## Bài 2: Dãy Fibonacci

**Fibonacci**( $n$ )

1     **if**  $n \leq 0$  **return** 0

2     **if**  $n = 1$  **return** 1

3     **return** **Fibonacci**( $n - 1$ )  
          + **Fibonacci**( $n - 2$ )

Thuật toán đệ quy:  $\Omega(\phi^n)$  (thời gian hàm mũ),  
với  $\phi = (1 + \sqrt{5})/2$  – *golden ratio*





**Divide And Conquer  
có hoạt động tốt trong  
trường hợp này ?**

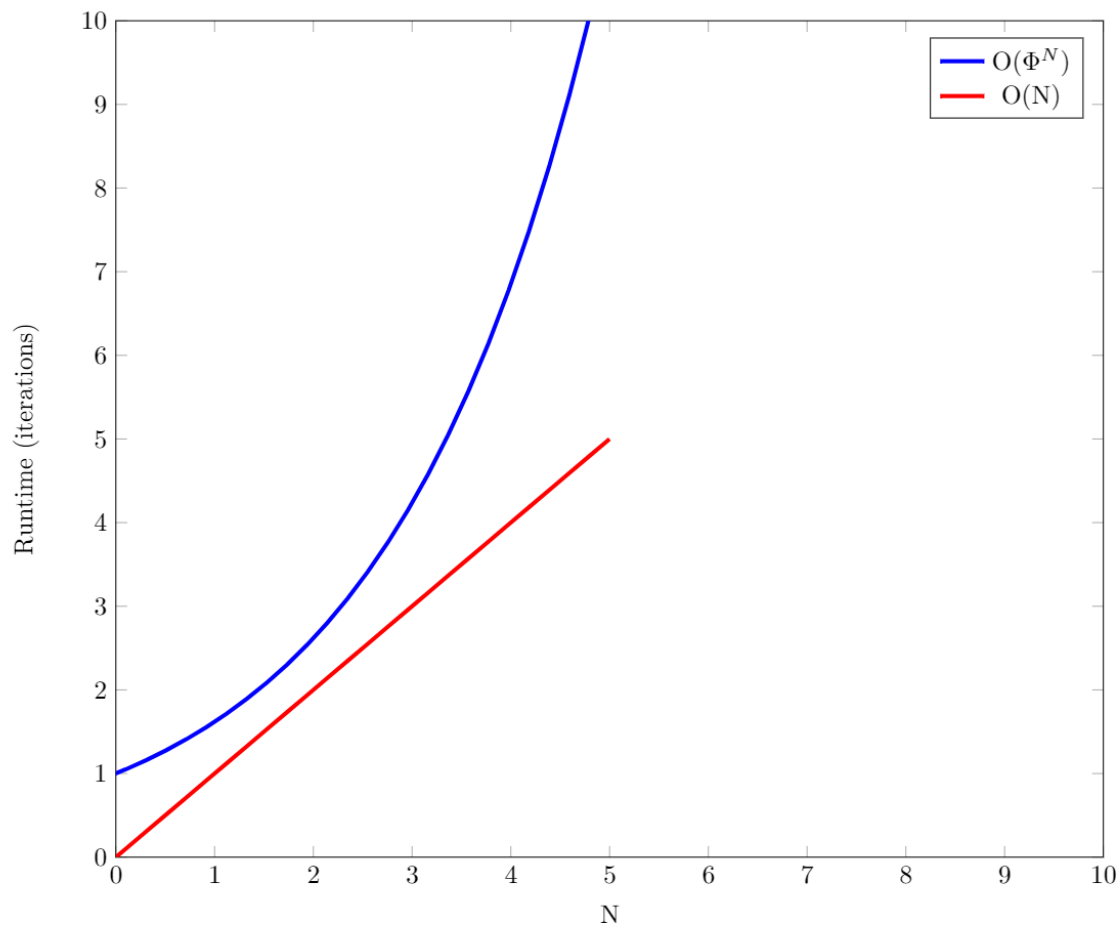
# 4

## Các phương pháp khác

- Dynamic Programming: Tính lần lượt  $F_0, F_1, F_2 \dots F_n$  theo thứ tự số sau bằng tổng hai số trước

```
def fib(n):  
    table = []  
    table.append(0)  
    table.append(1)  
    for i in range(2, n+1):  
        table.append(table[i-1] + table[i-2])  
    return(table[n])
```

$$\Rightarrow T(n) \in \theta(n)$$





# THANKS!

**Any questions?**