



THẢO LUẬN ĐỒ ÁN MÔN HỌC:

PHÂN TÍCH & THIẾT KẾ

THUẬT TOÁN

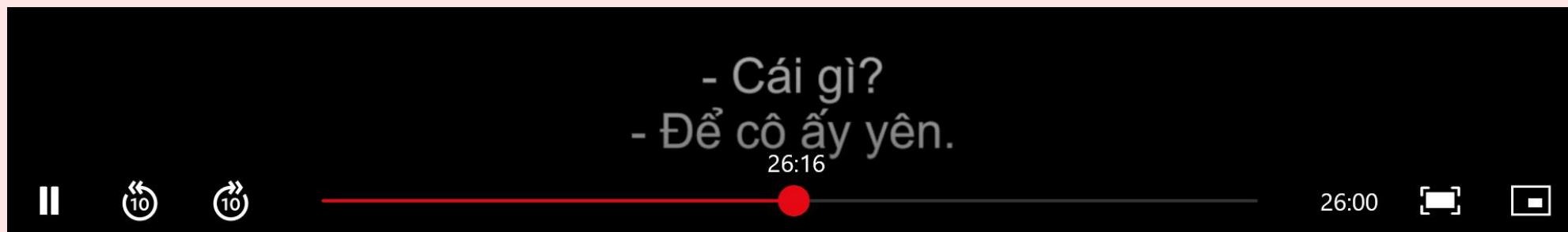
Hello ~

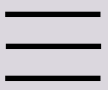


NHÓM | 19522065 – Nguyễn Thị Minh Phương
01 | 19522065 – Phan Nhật Minh



ĐẶT VẤN ĐỀ





CHỦ ĐỀ

CHIA ĐỀ TRÌ!

DIVIDE and CONQUER





- 1** Đặt vấn đề
- 2** Khái niệm
- 3** Các bước thực hiện
- 4** Phân tích độ phức tạp thuật toán
- 5** Các bài toán áp dụng
- 6** Ưu điểm & nhược điểm
- 7** Quizzes
- 8** Tổng kết & BTVN
- 9** Tài liệu tham khảo

TABLE OF CONTENTS



Khái niệm



Chia để trị

Là phương pháp giải quyết một bài toán bằng cách *chia nhỏ* nó ra thành các bài toán con, ta sẽ tìm cách *giải quyết* các bài toán con này. Sau đó lời giải của các bài toán con được *tổng hợp* lại thành lời giải cho bài toán ban đầu.



**Các bước
thực hiện**



01

Divide.

—

Chia bài toán ban đầu thành
nhiều bài toán con nhỏ hơn.



**Các bước
thực hiện**



01

Divide.

Chia bài toán ban đầu thành nhiều bài toán con nhỏ hơn.

02

Conquer.

Tìm phương án để giải quyết từng bài toán con một cách cụ thể.



01

Divide.

Chia bài toán ban đầu thành nhiều bài toán con nhỏ hơn.

02

Conquer.

Tìm phương án để giải quyết từng bài toán con một cách cụ thể.

03

Combine.

Kết hợp lời giải của các bài toán con để suy ra kết quả của bài toán ban đầu cần tìm.

**Các bước
thực hiện**



01

Divide.

Chia bài toán ban đầu thành nhiều bài toán con nhỏ hơn.

02

Conquer.

Tìm phương án để giải quyết từng bài toán con một cách cụ thể.

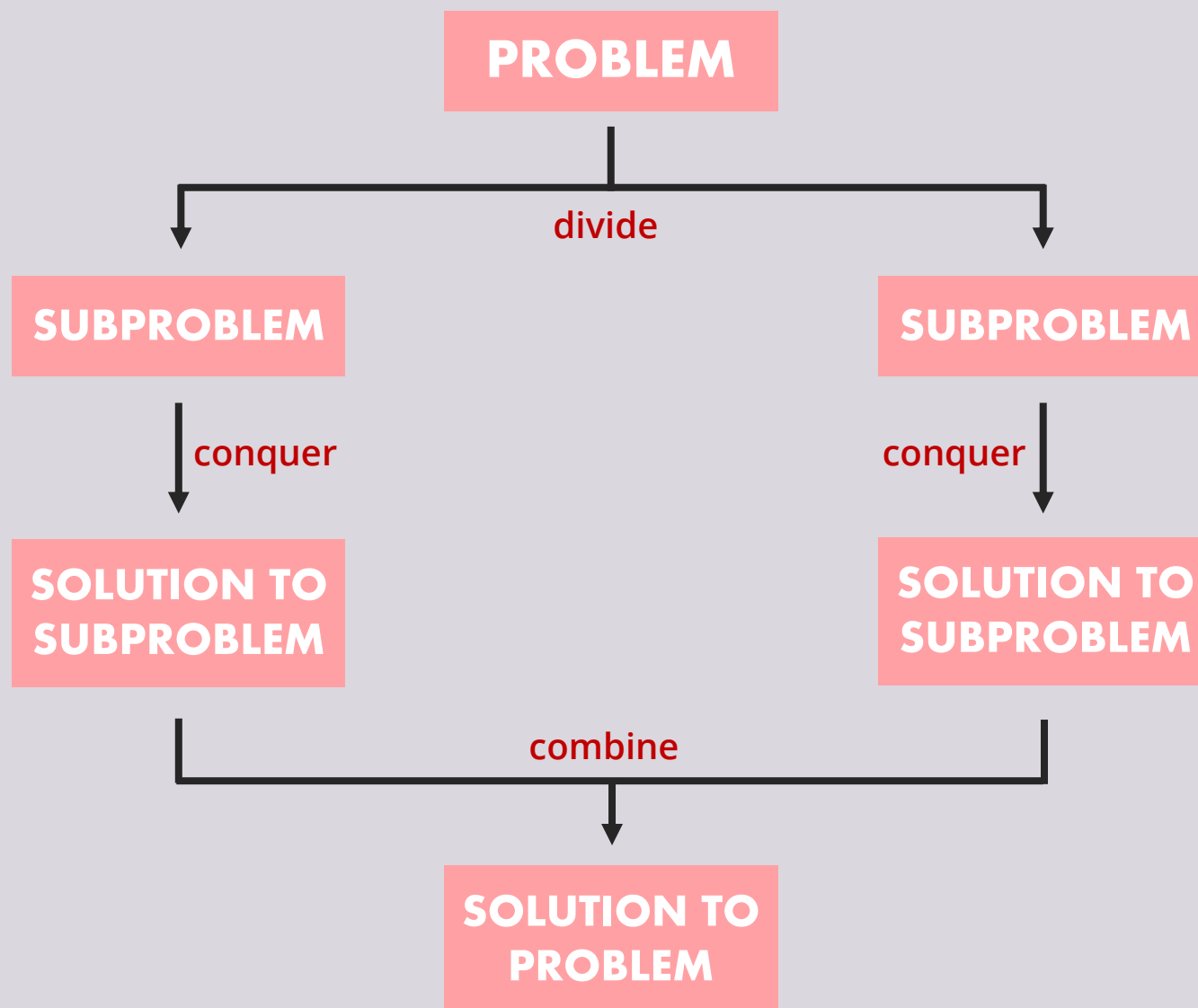
03

Combine.

Kết hợp lời giải của các bài toán con để suy ra kết quả của bài toán ban đầu cần tìm.

**Các bước
thực hiện**





```
DAC(P) {  
    if(small(P)) {  
        Solution(P);  
    }  
    else {  
        divide P into P1, P2,..., Pk  
        apply DAC(P1), DAC(P2),..., DAC(Pk)  
        combine(DAC(P1), DAC(P2),..., DAC(Pk))  
    }  
}
```



Master Theorem

If $f(n) \in \Theta(n^d)$ where $d \geq 0$ in recurrence

$$T(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d, \\ \Theta(n^d \log n) & \text{if } a = b^d, \\ \Theta(n^{\log_b a}) & \text{if } a > b^d. \end{cases}$$



Master Theorem

If $f(n) \in \Theta(n^d)$ where $d \geq 0$ in recurrence

$$T(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d, \\ \Theta(n^d \log n) & \text{if } a = b^d, \\ \Theta(n^{\log_b a}) & \text{if } a > b^d. \end{cases}$$

Examples

$$T(n) \in \Theta(?)$$

1. $T(n) = 4T(\frac{n}{2}) + n$

$$d = 1, a = 4, b = 2 \Rightarrow a > b^d (4 > 2^1) \Rightarrow T(n) \in \Theta(n^2)$$

2. $T(n) = 4T(\frac{n}{2}) + n^2$

$$d = 2, a = 4, b = 2 \Rightarrow a = b^d (4 = 2^2) \Rightarrow T(n) \in \Theta(n^2 \log(n))$$

$$T(n) = 4T\left(\frac{n}{2}\right) + n^3$$

$$d = 3, a = 4, b = 2 \Rightarrow a < b^d (4 < 2^3) \Rightarrow T(n) \in \Theta(n^3)$$

Bài tập áp dụng



Problem

Cho 2 số nguyên không âm a, b với $a \neq 0, 10^5 \leq b \leq 10^{18}$.

Tính $a^b \bmod (10^9 + 7)$.

Solution

$$a^b = \begin{cases} a^{b/2} * a^{b/2} & (\text{nếu } b \text{ chẵn}) \\ a^{b/2} * a^{b/2} * a & (\text{nếu } b \text{ lẻ}) \end{cases}$$

$$\begin{aligned} a^b &= a^{b/2} * a^{b/2} \\ &= a^{b/4} * a^{b/4} * a^{b/4} * a^{b/4} \\ &= \dots \\ &= \underbrace{a * a * \dots * a}_{b\text{-times}} \end{aligned}$$

Base case

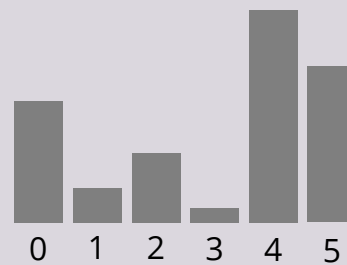
$$b = 0$$

Code

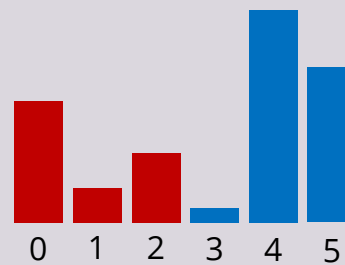
[Code minh họa](#)



MERGE SORT



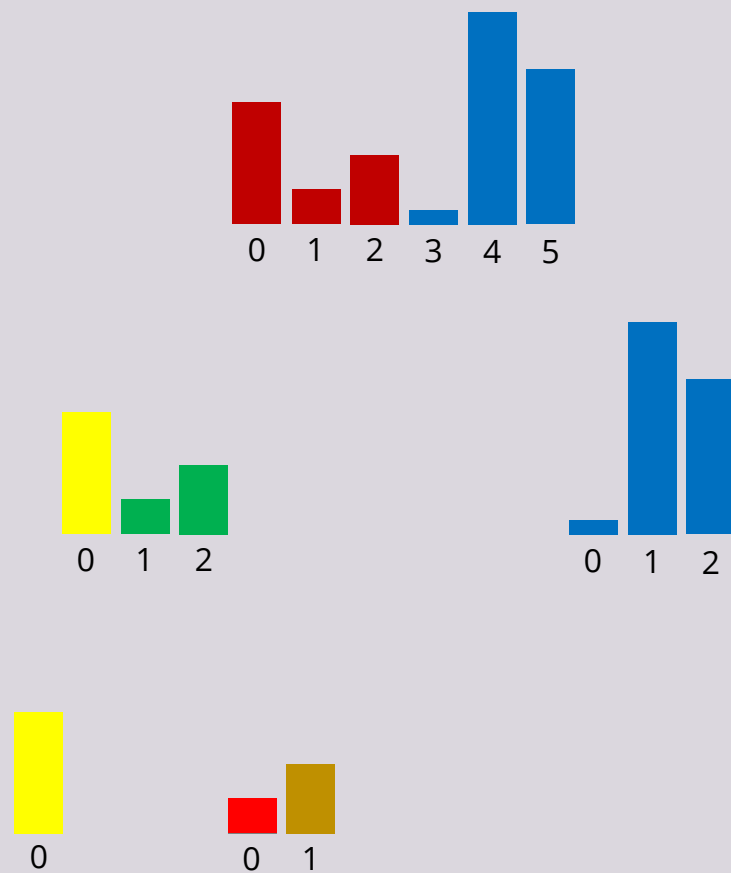
MERGE SORT



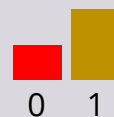
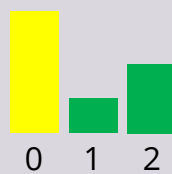








Number of Children	Frequency
0	4
1	2
2	3
3	1
4	6
5	5













The figure displays three bar charts, each representing a different category. The top chart has six bars for categories 0 through 5, with bars 0-3 in red and bars 4-5 in blue. The bottom-left chart has three bars for categories 0 through 2, all in red. The bottom-right chart has three bars for categories 0 through 2, with bar 0 in yellow and bars 1-2 in green.

Category	Count	Color
0	3	Red
1	1	Red
2	2	Red
3	0.5	Blue
4	4	Blue
5	3.5	Blue

Category	Count	Color
0	1	Red
1	2	Red
2	3	Red

Category	Count	Color
0	0.5	Yellow
1	4	Green
2	3.5	Green











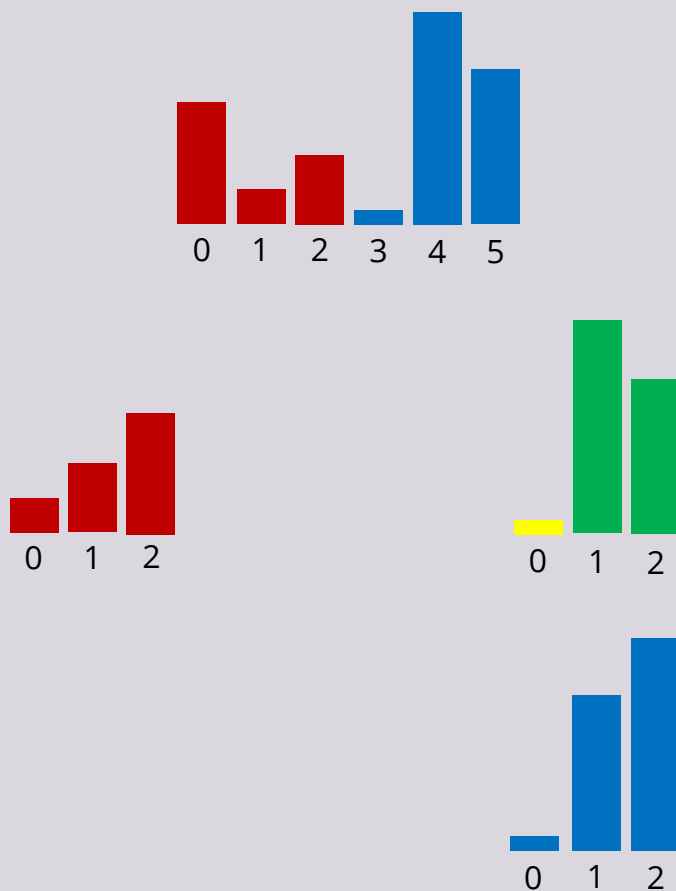
The figure displays a 3x3 grid of bar charts, each representing a different distribution of a variable across three categories (0, 1, 2). The distributions are as follows:

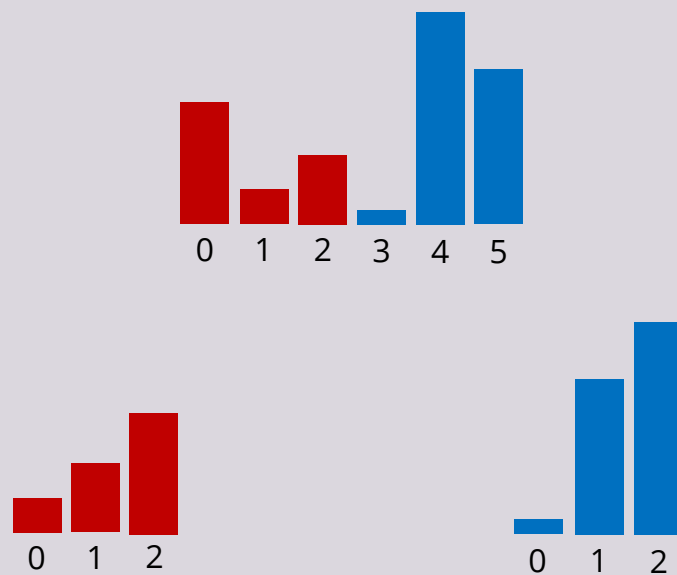
- Top Row:** Red bars for categories 0, 1, and 2; Blue bars for categories 3, 4, and 5.
- Middle Row:** Red bars for categories 0, 1, and 2; Green bars for categories 1 and 2; Yellow bar for category 0.
- Bottom Row:** Red bar for category 0; Purple bar for category 1; Green bars for categories 0 and 1.

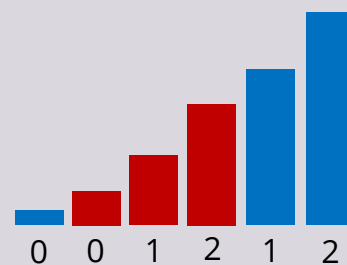






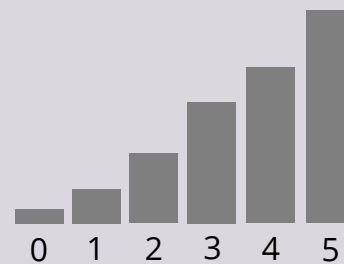




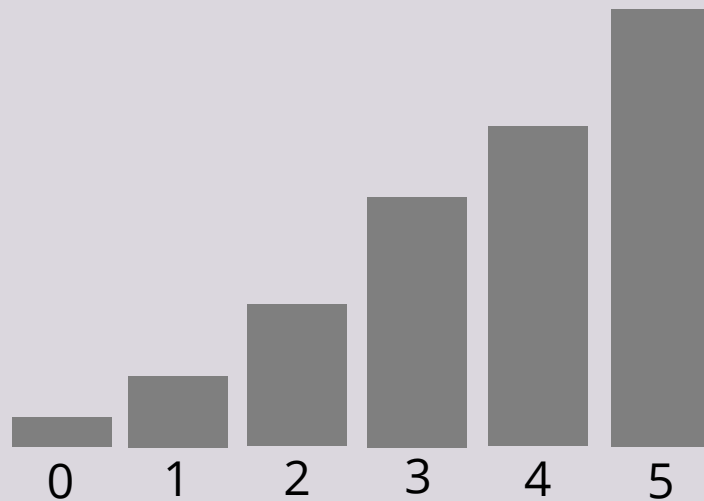




MERGE SORT



MERGE SORT



$$T(n) \in \Theta(n \log(n))$$

Code

[Code minh họa](#)



Karatsuba's Algorithm

Multiplication of Large Integers

Ví dụ: 23×14

$$23 = 2.10^1 + 3.10^0$$

$$14 = 1.10^1 + 4.10^0$$

$$\begin{aligned} 23 * 14 &= (2.10^1 + 3.10^0) * (1.10^1 + 4.10^0) \\ &= (2 * 1).10^2 + (2 * 4 + 3 * 1).10^1 + (3 * 4).10^0 \end{aligned}$$

4 phép nhân

$$2 * 4 + 3 * 1 = (2 + 3) * (1 + 4) - 2 * 1 - 3 * 4$$

$$\begin{aligned} 23 * 14 &= (2 \cdot 10^1 + 3 \cdot 10^0) * (1 \cdot 10^1 + 4 \cdot 10^0) \\ &= (2 * 1) \cdot 10^2 + ((2 + 3) * (1 + 4) - 2 * 1 - 3 * 4) \cdot 10^1 + (3 * 4) \cdot 10^0 \end{aligned}$$

3 phép nhân

Karatsuba's Algorithm

Multiplication of Large Integers

Recurrence Relation

$$A(n) = 3A(n/2) + cn \quad \text{for } n > 1, \quad A(1) = 1.$$

Độ phức tạp

$$A(n) \in \Theta(n^{\log_2 3})$$

Code

[Code minh họa](#)



MỘT SỐ BÀI TOÁN KHÁC:

- Binary Search
- Quick Sort
- Binary Tree Traversals and Related Properties
- Strassen's Matrix Multiplication
- The Closest-Pair and Convex-Hull Problems by Divide-and-Conquer



Ưu điểm & nhược điểm của kỹ thuật chia để trị

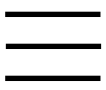
Ưu điểm

- Giải quyết được bài toán lớn, phức tạp.
- Trực quan, dễ cài đặt.
- Độ phức tạp thường thấp hơn Brute Force.
- Tìm ra giải thuật hiệu quả.
- Sử dụng bộ nhớ đệm hiệu quả.
- Độ chính xác cao.
- Giải quyết các vấn đề con song song bằng đa luồng.

Nhược điểm

- Độ quy thực thi chậm, dễ xảy ra lỗi khi cài đặt.
- Tốn thêm bộ nhớ.
- Khó khăn trong việc chọn điều kiện dừng.
- Trùng lặp vấn đề con khi chia nhỏ.
- Đôi khi sẽ phức tạp.

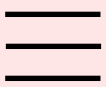




Quizzes

- Mỗi nhóm 1 đội
- Nickname: nhóm01, nhóm02,...





TỔNG KẾT

Bài tập

Tìm vị trí của phần tử lớn nhất trong mảng gồm n phần tử, sử dụng kỹ thuật chia để trị.

1. Viết chương trình thực hiện bài toán.
2. Giải recurrence relation của bài toán.
3. So sánh với thuật toán Brute Force.

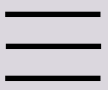




Tài liệu tham khảo

- Anany Levition, Introduction to the Design and Analysis of Algorithms, 3rd edition, 2014
- https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm





THANKS FOR LISTENING



Trả lời các câu hỏi:

Bài toán nào có thể áp dụng kỹ thuật chia để trị?

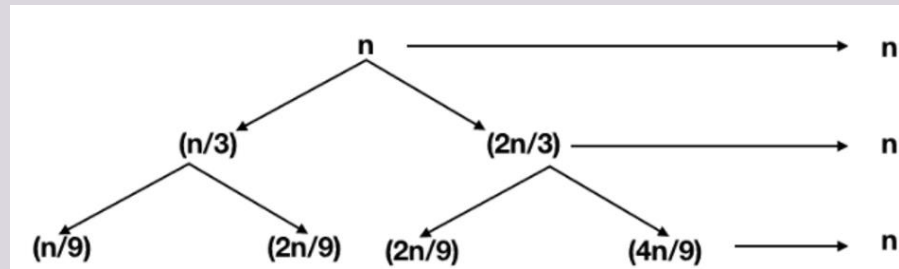
Những bài toán lớn có thể chia ra thành những bài toán con *độc lập, không gối lên nhau*.

Làm sao để tính độ phức tạp khi chia bài toán ra thành các bài toán con không đều nhau?

Sử dụng phương pháp cây đệ quy.

Ví dụ:

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$



Sự khác nhau giữa a và b trong công thức truy hồi tổng quát (Recurrence Relation)?

$$T(n) = a * T\left(\frac{n}{b}\right) + f(n)$$

b: tổng số vấn đề con sau khi chia nhỏ

a: số vấn đề con *cần được giải quyết*

Ví dụ:

$$a^b = a^{b/2} * a^{b/2}$$

Ở đây $b=2$ và $a=1$ vì ta chỉ cần tính $a^{\frac{b}{2}}$ một lần.