DIVIDE

CONQUER

WHAT?

Chia để trị là gì

WHAT?

Là một chiến lược thiết kế thuật toán:

- Top-down approach
- Thường được cài đặt bằng đệ quy
- Chia - trị - hợp

Ý tưởng cơ bản

Gồm 3 phần:

- Chia: chia bài toán lớn thành 2 hoặc nhiều bài toán con nhỏ hơn để giải quyết
- Trị: Giải quyết các bài toán con
- Hợp: Kết hợp kết quả của các bài toán con để có được lời giải cuối cùng cho bài bài toán lớn ban đầu

Các bước của chia để trị?

Tùy bài toán

Không dùng đệ quy được không?

Được

Tính chất của bài toán con?

Đồng dạng với bài toán lớn

WHY?

Điểm mạnh/Điểm yếu

WHEN?

Khi nào áp dụng chia để trị

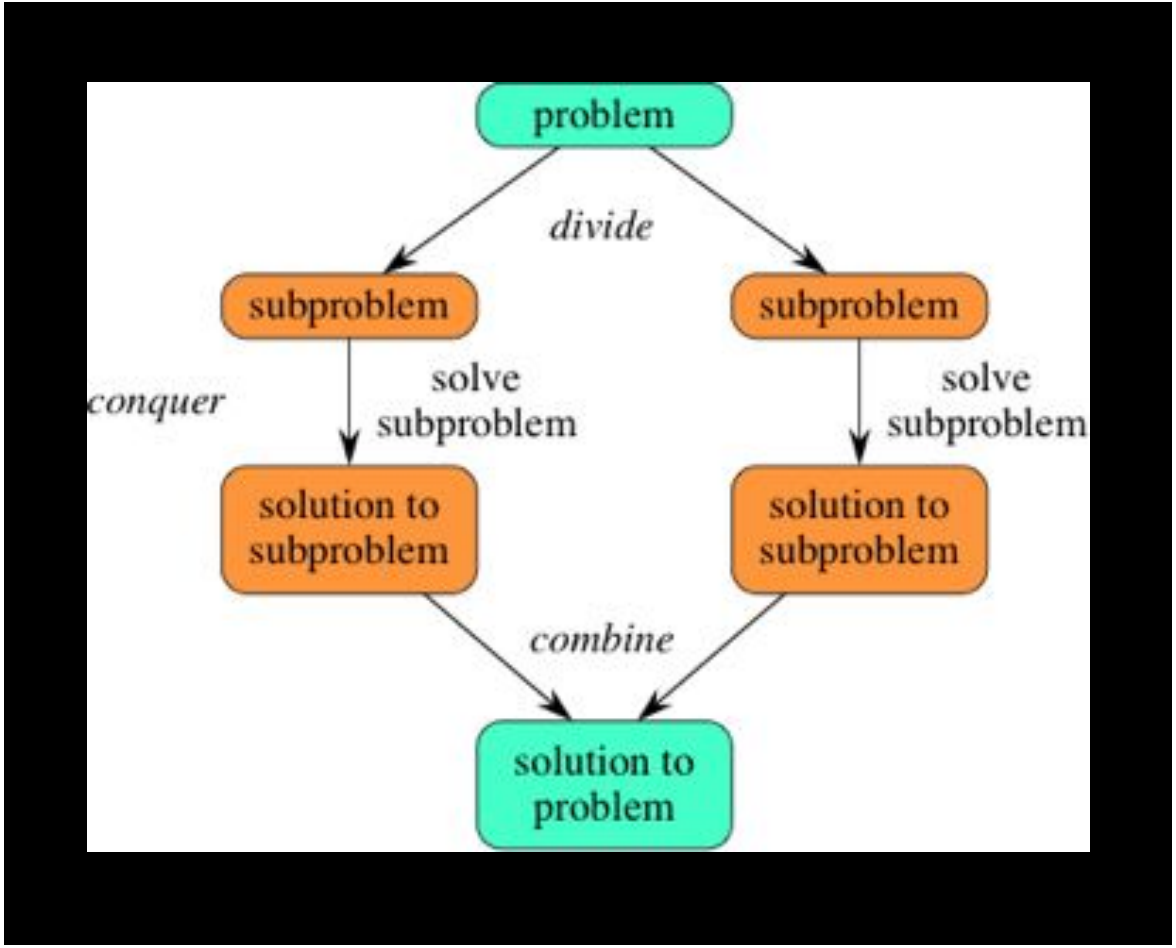
WHERE?

Ứng dụng vào đâu

HOW?

Cài đặt thế nào

FLOWCHART


problem
divide
subproblem
subproblem
conquer
solve subproblem
solve subproblem
solution to subproblem
solution to subproblem
combine
solution to problem

```
DnC (P) {
  If (P is small) {
    Solve(P)
  } else {
    Divide P into n smaller
sub-problem ρ
    Apply DnC(ρ_n)
    Combine(DAC(ρ_n))
  }
}
```

FRAMEWORK

Thời gian chạy

$$T(n) = a\ T(n/b) + f(n).$$

Với:

- ***T(n)*** *là thời gian chạy*
- *a là số vòng lặp đệ quy*
- ***T(n/b)*** *là thời gian chạy mỗi vòng lặp đệ quy*
- ***f(n)*** *là thời gian phân chia và hợp các bài toán con*

Một số dạng bài toán của chia để trị

- Merge Sort
- Quick Sort
- Binary Tree Traversals and Related Properties
- Calculate pow(x, n)

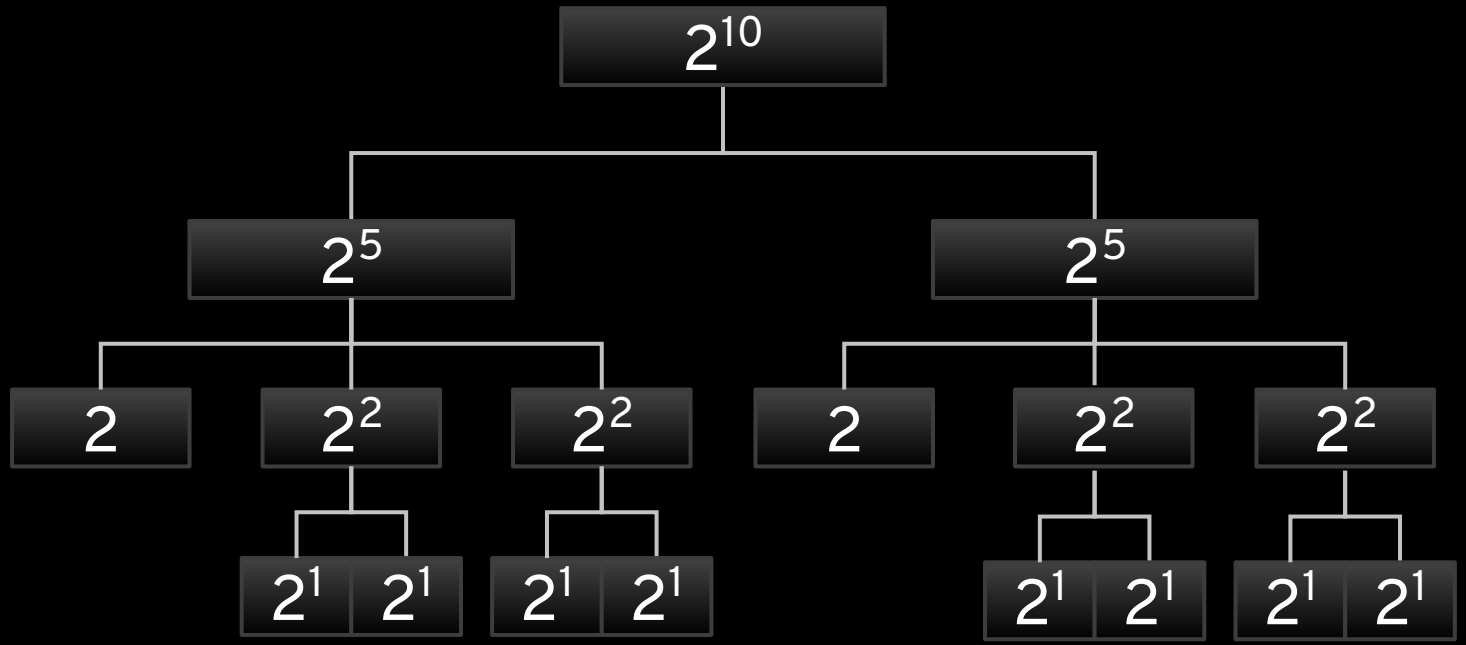
Một số dạng bài toán của chia để trị

- Multiplication of Large Integers
- Strassen’s Matrix Multiplication
- The Closest-Pair Problem
- Convex Hull (Simple Divide and Conquer Algorithm)

Tính pow(x, n)

2^{10}

Tính pow(x, n)



Tính pow(x, n)

Vấn đề có thể được xác định đệ quy bởi:

- power(x, n) = power(x, n / 2) * power(x, n / 2); // if n is even
- power(x, n) = x * power(x, n / 2) * power(x, n / 2);// if n is odd

```
def power(x, y):
    if (y == 0):
        return 1
    elif (int(y % 2) ==0):
        return (power(x, int(y / 2)) * power(x, int(y / 2)))
    else:
        return (x * power(x, int(y / 2)) * power(x, int(y / 2)))
```

```
def power(x,  y):
    if y == 0:
        return 1
    temp = power(x, int(y / 2))
    if y % 2 == 0:
        return temp * temp
    else:
        return x * temp * temp
```