

## I. Đề tài:

**Thiết kế thuật toán theo phương pháp chia để trị để tính  $a^x$  (với  $a$  và  $x$  là các số nguyên rất lớn) với 2 phiên bản: đệ qui và không đệ qui**

- **Input:** 2 số nguyên  $a, x$  (rất lớn)
- **Output:** kết quả của  $a^x$
- **Pattern recognition:**
  - $a^x = a * a * a * a * \dots$  (n lần), mỗi lần lặp ta thực hiện 1 phép nhân
- **Độ phức tạp thuật toán:**  $O(\log(x))$

## II. Thiết kế thuật toán

- **Ý tưởng thuật toán:** Ta sử dụng thuật toán chia để trị (Divide and Conquer) để xử lý những bước giống nhau
- **Ví dụ:**
  - $2^{13} = 2 \times 2^6 \times 2^6$
  - $2^6 = 2^3 \times 2^3$
  - $2^3 = 2 \times 2^1 \times 2^1$
  - $2^1 = 2 \times 2^0 \times 2^0$
- ➔ **Có 2 trường hợp để xử lý:**
  - **Nếu  $x$  là số lẻ:**  $a^x = a * a^{x/2} * a^{x/2}$
  - **Nếu  $x$  là số chẵn:**  $a^x = a^{x/2} * a^{x/2}$

- **Mã giả thuật toán - không đệ qui (pseudocode):**

```
def pow(a, x):  
    res ← 1                # a^0 base case  
    while x > 0:  
        if x is even:      # if x is even, add to the answer  
            res = res * a  
  
        x ← x / 2           # divide y by 2  
        a ← a * a           # multiply for the same  
  
    return res
```

### Code hoàn chỉnh (python):

```
def pow(a, x):
    res = 1

    while (x > 0):
        if (x % 2 == 1):
            res = res * a
        x = x >> 1
        a = a * a

    return res
```

### - Mã giả thuật toán - đệ qui (pseudocode):

```
def pow(a, x):
    if x == 0: return 1      # Base case:  $a^0 = 1$ 
    if x is odd:
        return a * pow(a, x/2) * pow(a, x/2)
    if x is even:
        return pow(a, x/2) * pow(a, x/2)
```

### Code hoàn chỉnh (python):

```
def pow(a, x):
    res = 0
    if x == 0:
        return 1
    res = pow(a, int(x / 2))
    if (x % 2):
        return a * res * res
    else: return res * res
```