

- Khái niệm
- Đặc điểm và ý nghĩa sử dụng
- Ví dụ minh họa

Hàm đệ quy



Khái niệm đệ quy

- Một hàm đệ quy là hàm gọi lại chính bản thân nó một cách trực tiếp hoặc gián tiếp thông qua các hàm khác
- Đệ quy là một chủ đề phức tạp sẽ được trình bày cụ thể trong môn cấu trúc dữ liệu và giải thuật
- Nội dung bài học này chỉ đề cập đến hàm đệ quy ở khía cạnh đơn giản nhất
- Ví dụ:

```
unsigned long long factorial(unsigned int n) {  
    if (n < 2) { // trường hợp cơ sở  
        return 1;  
    }  
    else { // bước đệ quy  
        return n * factorial(n - 1);  
    }  
}
```

Đặc điểm hàm đệ quy

- Hàm đệ quy thường chỉ biết cách xử lý vấn đề ở trường hợp đơn giản nhất hay còn gọi là trường hợp cơ sở, điểm dừng
- Với các lời gọi hàm trùng với trường hợp trên thì hàm đơn giản chỉ return giá trị cơ sở tương ứng
- Ví dụ:

```
unsigned long long factorial(unsigned int n) {  
    if (n < 2) { // trường hợp cơ sở  
        return 1;  
    }  
    else { // bước đệ quy  
        return n * factorial(n - 1);  
    }  
}
```

Đặc điểm hàm đệ quy

- Với các lời gọi phức tạp, hàm đệ quy thường chia vấn đề ra làm 2 phần, phần thứ nhất hàm này tự biết cách xử lý, phần thứ hai hàm không biết cách xử lý nhưng vì vấn đề lúc này có vẻ giống với vấn đề ban đầu nhưng nhỏ hơn một chút nên hàm sẽ gọi lại chính nó để tiếp tục giải quyết
- Hành động gọi lại chính nó được gọi là bước đệ quy hay lời gọi đệ quy
- Ví dụ:

```
unsigned long long factorial(unsigned int n) {  
    if (n < 2) { // truong hop co so  
        return 1;  
    }  
    else { // buoc de quy  
        return n * factorial(n - 1);  
    }  
}
```

Ý nghĩa sử dụng

- Dùng hàm đệ quy để giải quyết các vấn đề có tính chất truy hồi
- Mọi vấn đề có thể giải quyết bằng đệ quy thì đều có thể giải quyết bằng vòng lặp
- Tuy nhiên cách giải quyết bằng đệ quy luôn tường minh gọn gàng hơn
- Ví dụ: tính $n!$, tìm số Fibonacci thứ n , bài toán đặt hậu, tháp Hà Nội, mã đi tuần

Lưu ý

- Mỗi cách giải quyết vấn đề luôn có ưu điểm và nhược điểm riêng
- Đề quy có ưu điểm là ngắn gọn, tường minh
- Nhược điểm là tốn thời gian và bộ nhớ
- Để đảm bảo vấn đề có thể giải quyết bằng đệ quy thì sau mỗi bước đệ quy phải giảm dần độ phức tạp về trường hợp cơ sở

Ví dụ minh họa

Định nghĩa đệ quy của $n!$ là $n! = n * (n - 1)!$, cụ thể:

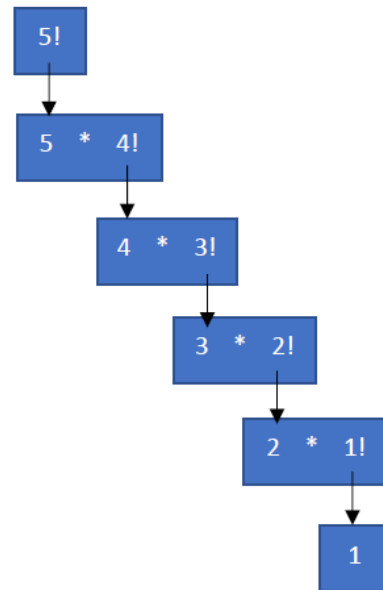
$$5! = 5 * 4 * 3 * 2 * 1$$

$$5! = 5 * (4 * 3 * 2 * 1)$$

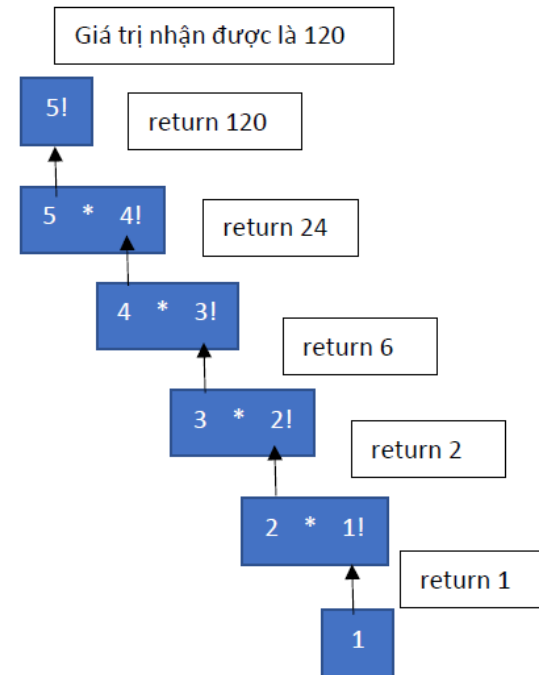
$$5! = 5 * (4!)$$

Phân tích chi tiết:

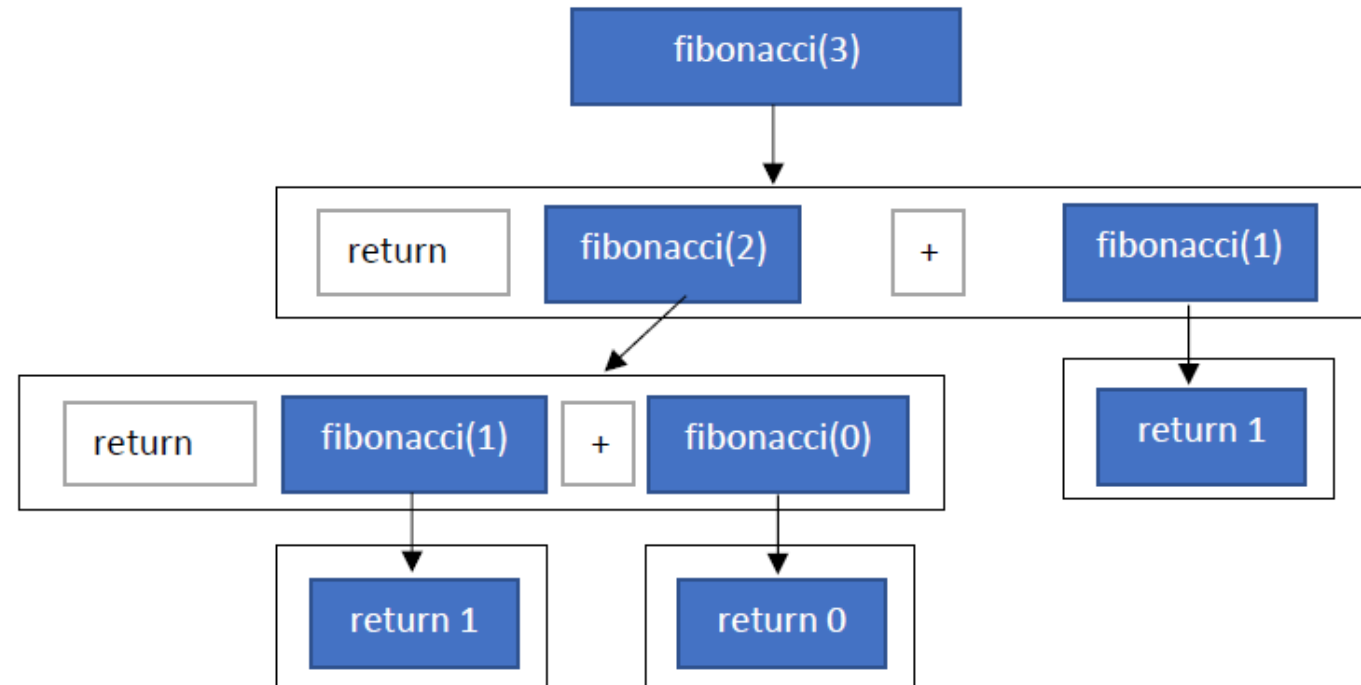
Lời gọi:



Giá trị trả về từ mỗi lời gọi đệ quy



Ví dụ minh họa



Tiếp theo

So sánh giữa đệ quy và vòng lặp

