START

Analysis recursive algorithm

Nhóm 13 - Phân tích độ phức tạp các thuật toán có đệ qui

TRẦN DUY TÂN 21522576

LÊ NHẬT MINH 21522339

TRẦN VĂN TOÁN 21522689

CS112 : Phân Tích Và Thiết Kế Thuật Toán

Mục Tiêu

01 biết cách tính độ phức tạp các thuật toán Đệ qui

02 Biết cách ước tính độ phức tạp của thuật toán đệ qui

03 Phương pháp phân tích độ phức tạp thuật toán đệ qui.

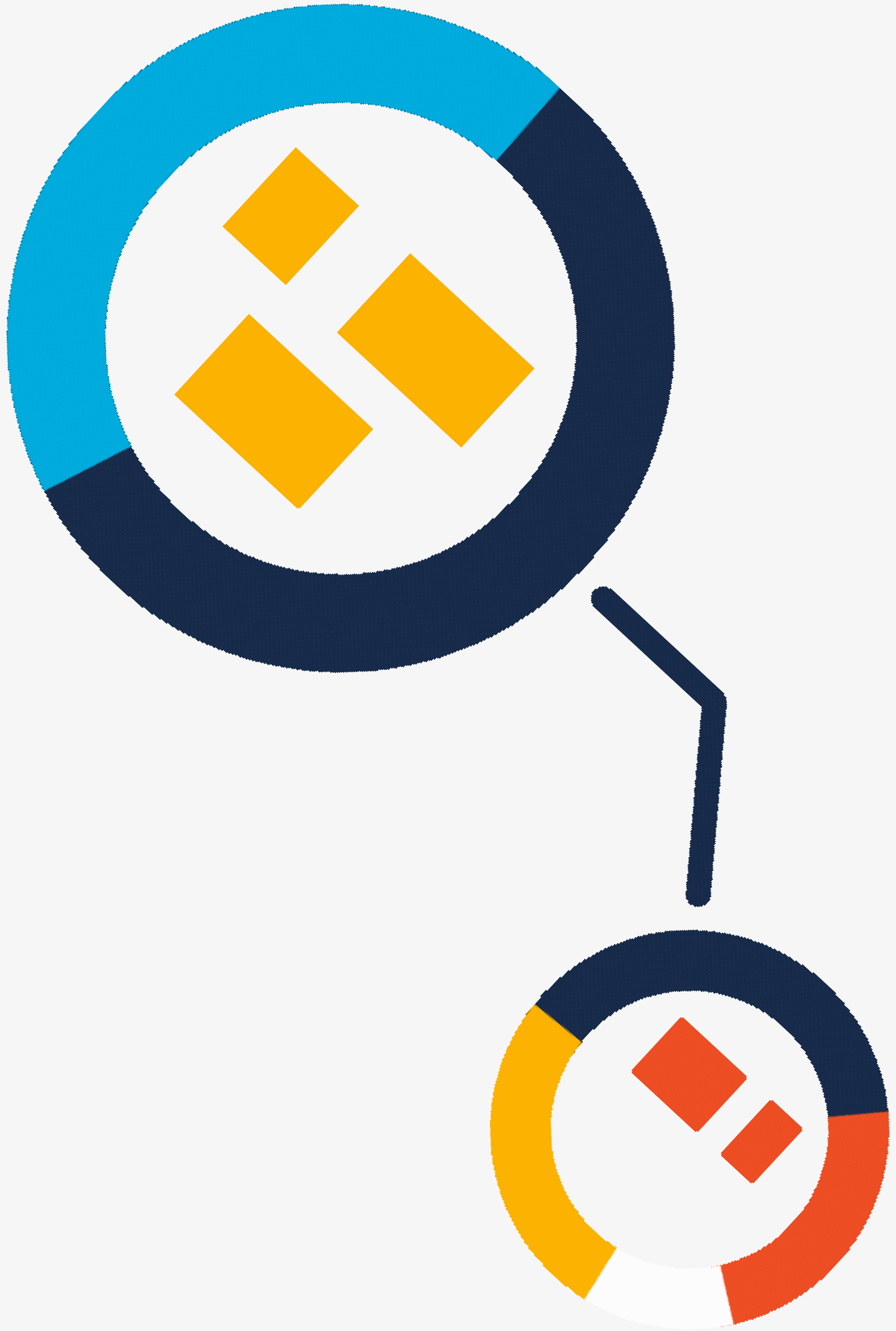
04 Mỗi nội dung có kèm theo ví dụ minh họa và bài tập

câu hỏi 1

Đệ quy là gì ?

câu hỏi 2

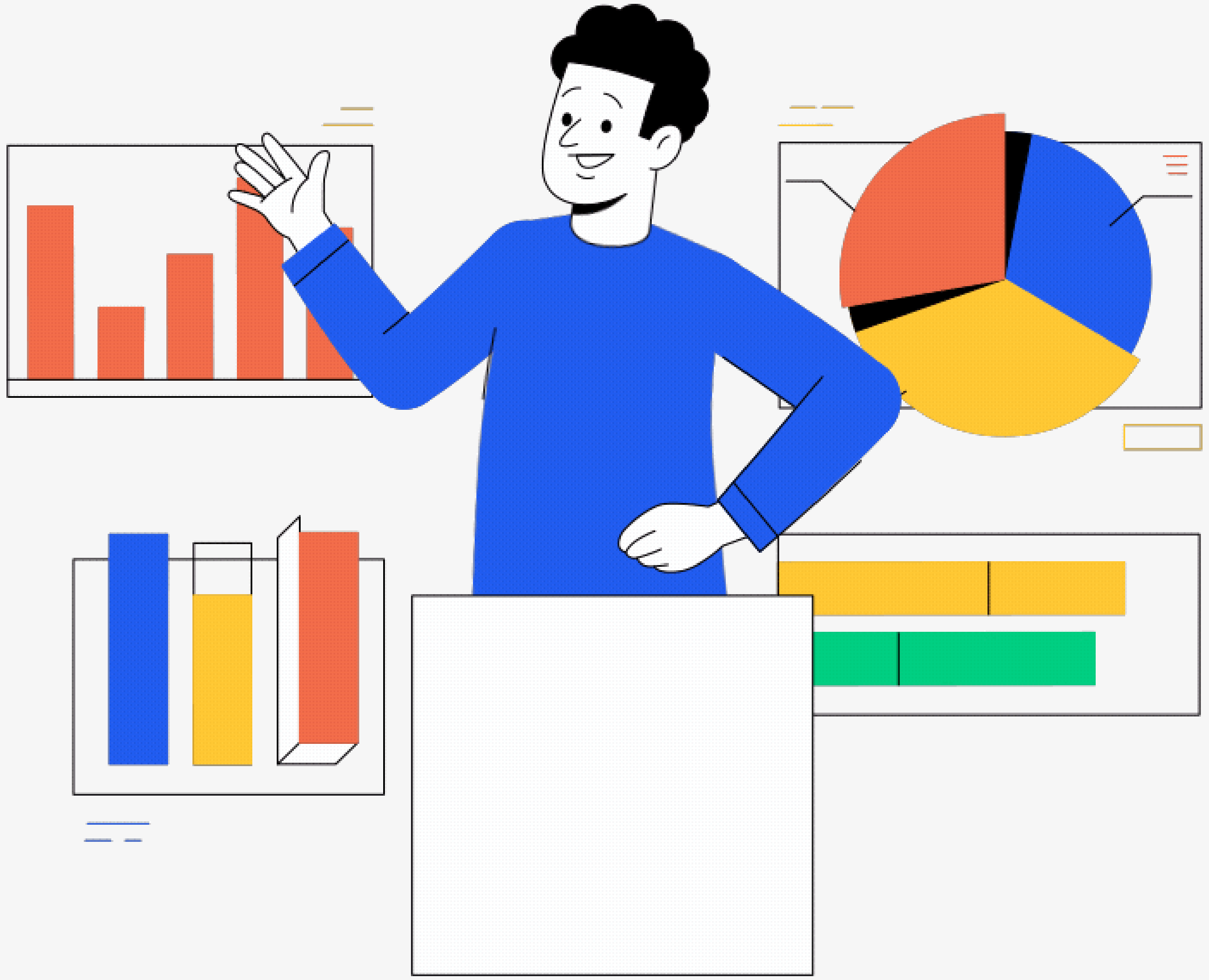
Các Thuật Toán Đã Học Có Áp Dụng Đệ Quy



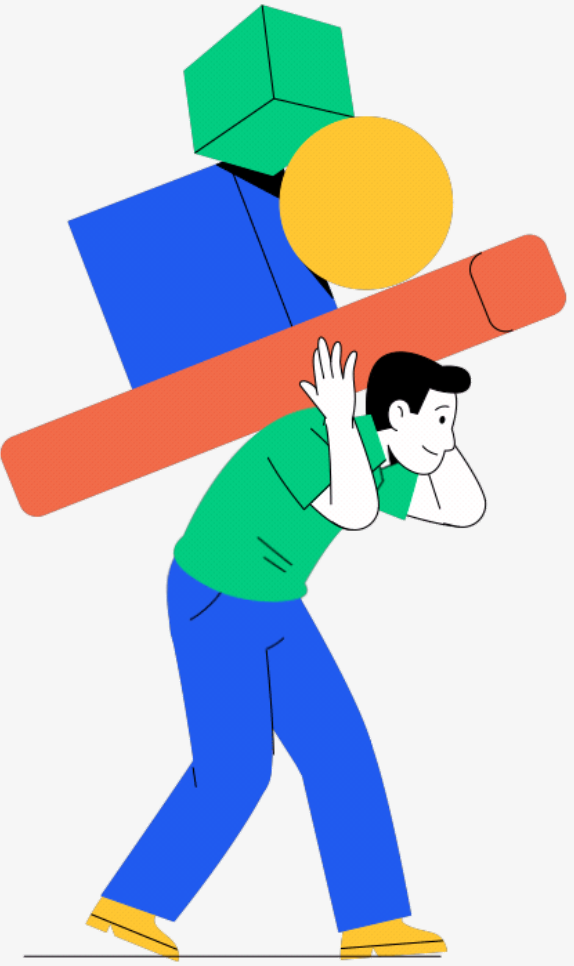
câu hỏi 3

Ưu điểm và nhược điểm của đệ quy và vòng lặp

câu hỏi 4



Thuật toán tính n!



1. Tham số đầu vào là n

2. Phép toán cơ bản là phép nhân

3. Quan hệ phụ thuộc : M(n) = M(n-1) + 1 với n > 0

4. Phương trình đệ quy : M(n) = M(n -1) +1 với n > 0
M(0) = 0

5. Giải phương trình đệ quy : M(n) = n.
độ phức tạp : n

Các bước xác định độ phức tạp của thuật toán đệ quy

1. Xác định tham số đầu vào

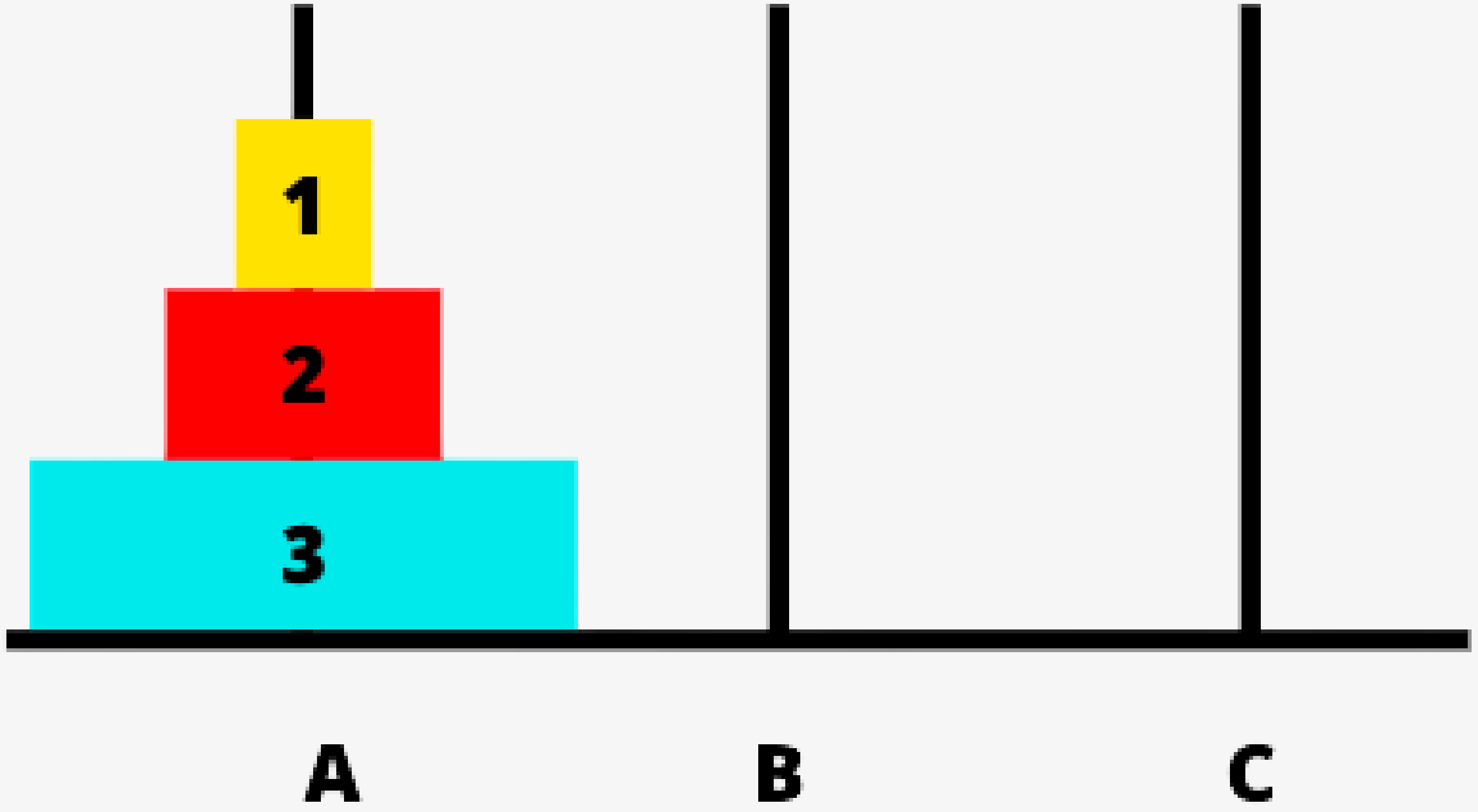
2. Xác định phép toán cơ bản của thuật toán

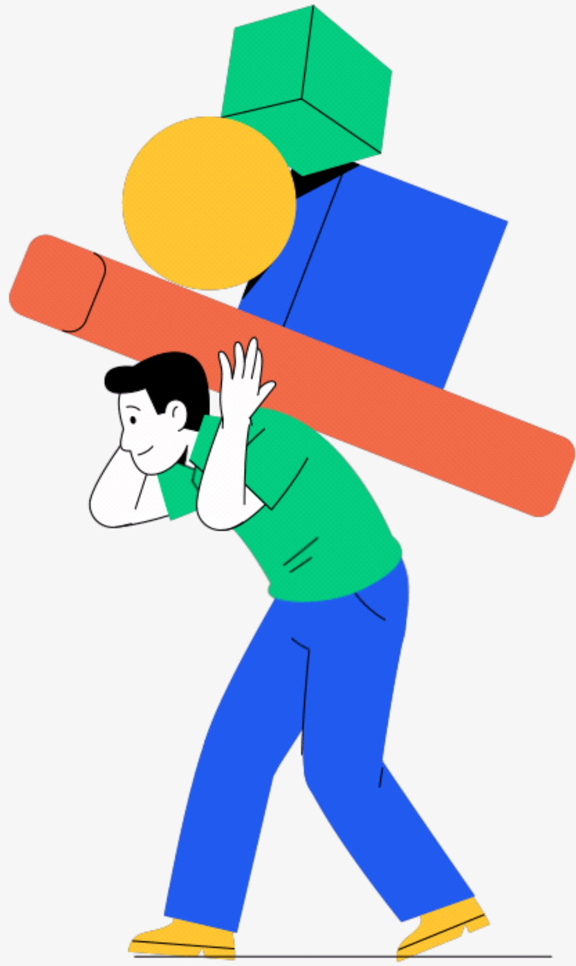
3. Xác định quan hệ phụ thuộc của hàm số vào cái tham số đầu vào

4. Thiết lập hệ thức truy hồi (phương trình đệ quy) với điều kiện khởi tạo cho số lần thực thi của phép toán cơ bản

5. Giải phương trình đệ quy,ước tính độ phức tạp của thuật toán

Đề : Có n đĩa với kích thước khác nhau và 3 cọc A, B, C. Ban đầu tất cả các đĩa đều ở cọc A, với thứ tự đĩa có kích thước lớn nhất nằm ở dưới cùng và nhỏ dần đến đĩa nhỏ nhất ở trên cùng . Mục đích là di chuyển tất cả các đĩa từ cột A đến cột C, di chuyển 1 đĩa mỗi lần sao cho đĩa lớn hơn luôn nằm ở dưới và trên nó là các đĩa bé hơn.



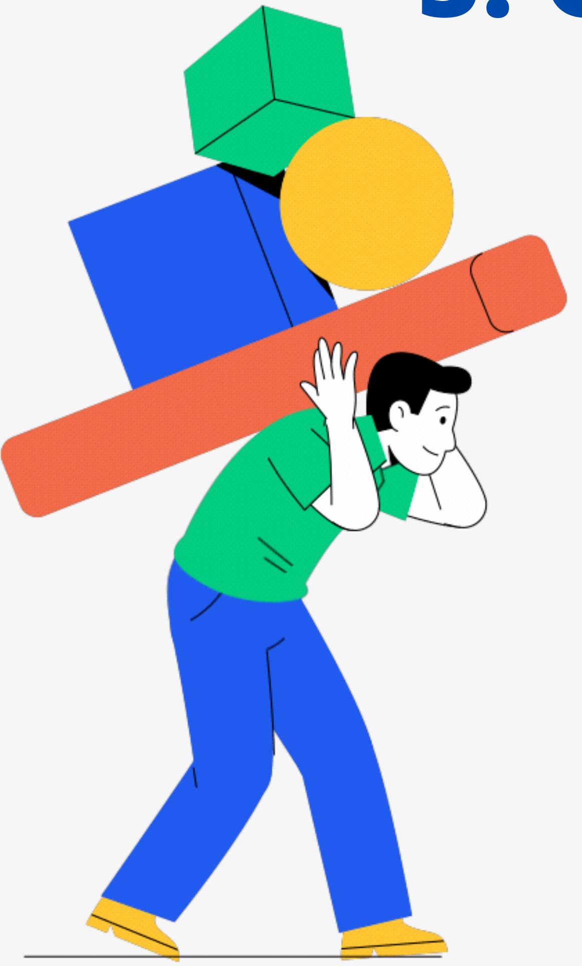


1. Tham số đầu vào là n
2. Phép toán cơ bản là phép "di chuyển"
3. Quan hệ phụ thuộc : $M(n) = M(n-1) + 1 + M(n-1)$ với $n > 1$
4. Phương trình đệ quy : $M(n) = 2M(n-1) + 1$ với $n > 1$

   $M(1) = 1$

5. Giải phương trình đệ quy : $M(n) = 2 ^ n - 1$

   độ phức tạp : $2 ^ n$



Phương pháp thay thế tịnh tiến(Forward Substitutions)

Xét truy hồi sau:

$$x(n) = 2x(n-1) + 1 \quad \text{với } n > 1,$$
$$x(1) = 1.$$

Ta thu được một số số hạn đầu như sau:

$$x(1) = 1,$$
$$x(2) = 2x(1) + 1 = 2 \cdot 1 + 1 = 3,$$
$$x(3) = 2x(2) + 1 = 2 \cdot 3 + 1 = 7,$$
$$x(4) = 2x(3) + 1 = 2 \cdot 7 + 1 = 15.$$

Dễ dàng thấy quy luật của dãy số này, ta có kết quả là :

$$x(n) = 2^n - 1 \text{ với } n = 1, 2, 3, \text{ và } 4.$$

Phương pháp thay thế ngược(Backward Substitutions)

Dùng phương pháp backward substitution để giải truy hồi sau:
$x(n) = x(n-1) + n$. với x(0) = 0
$x(n) = [x(n-2) + n - 1] + n = x(n-2) + (n-1) + n.$
$x(n) = [x(n-3) + n - 2] + (n-1) + n = x(n-3) + (n-2) + (n-1) + n.$
Ta có thể nhận thấy quy luật sau:

$$x(n) = x(n-i) + (n-i+1) + (n-i+2) + \cdots + n.$$

vì điều kiện dừng xảy ra khi n=0, nên cần thực hiện i=n lần(giải n-i=0),
Suy ra:

$$x(n) = x(0) + 1 + 2 + \cdots + n = 0 + 1 + 2 + \cdots + n = n(n+1)/2.$$

Truy hồi bậc 2 với hệ số không đổi

Xét phương trình hồi quy:

$$ax(n) + bx(n-1) + cx(n-2) = f(n)$$

$f(n) = 0$ với mọi n, gọi là hồi quy đồng nhất.
Nếu ngược lại, gọi là hồi quy không đồng nhất.

Phương trình hồi quy đồng nhất
f(n) = 0 với mọi n

Xét trường hợp hồi quy đồng nhất:

$$ax(n) + bx(n-1) + cx(n-2) = 0 \quad (1)$$

Đặt $r^2 = x(n), \quad r = x(n-1), \quad x(n-2) = 1$
Ta thu được phương trình như sau:

$$ar^2 + br + c = 0 \quad (a)$$

Gọi r_1, r_2 là hai nghiệm của phương trình đặc trưng (a) cho phương trình hồi quy (*).

Trường hợp 1: Nếu r_1 và r_2 là các nghiệm thực và phân biệt, ngh quát cho phương trình truy hồi (1)

$$x(n) = \alpha r_1^n + \beta r_2^n$$

trong đó, α và β là hai hằng số thực chưa biết. Hai hằng số này bằng cách giải hệ phương trình lập từ các điều kiện dừng của phương

Trường hợp 2: Nếu $r_1 = r_2$(nghiệm kép), nghiệm tổng quát của phương trình truy hồi (1) có dạng như sau:

$$x(n) = \alpha r^n + \beta n r^n$$

Trong đó $r = r_1 = r_2$ và α và β là hai số thực chưa biết.

Trường hợp 3: Nếu $r_{1,2} = u \pm iv$ là hai nghiệm phức phân biệt, nghiệm tổng quát của phương trình hồi quy (1) có dạng:

$$x(n) = \gamma^n [\alpha \cos n\theta + \beta \sin n\theta]$$

Trong đó $\gamma = \sqrt{u^2 + v^2}, \theta = \arctan v/u$, α và β là hai số thực chưa biết.

Ví dụ: Tính độ phức tạp của thuật toán tính số Fibonacci thứ n, biết $F(0) = 0$ và $F(1) = 1$:

$$F(n) - F(n-1) - F(n-2) = 0.$$

Phương trình đặc trưng

$$r^2 - r - 1 = 0$$

các nghiệm vủa phương trình:

$$r_{1,2} = \frac{1 \pm \sqrt{1 - 4(-1)}}{2} = \frac{1 \pm \sqrt{5}}{2}$$

Do hai nghiệm phân biệt, nên ta có:

$$F(n) = \alpha \left(\frac{1 + \sqrt{5}}{2} \right)^n + \beta \left(\frac{1 - \sqrt{5}}{2} \right)^n .$$

Ta đã biết $F(0) = 0$ and $F(1) = 1$:

$$F(0) = \alpha \left(\frac{1+\sqrt{5}}{2}\right)^0 + \beta \left(\frac{1-\sqrt{5}}{2}\right)^0 = 0,$$
$$F(1) = \alpha \left(\frac{1+\sqrt{5}}{2}\right)^1 + \beta \left(\frac{1-\sqrt{5}}{2}\right)^1 = 1.$$

Giải hệ phương trình:

$$\begin{aligned} \alpha + \quad \beta &= 0 \\ \left(\frac{1+\sqrt{5}}{2}\right)\alpha + \left(\frac{1-\sqrt{5}}{2}\right)\beta &= 1. \end{aligned}$$

Nghiệm hệ phương trình: $\alpha = 1/\sqrt{5} \quad \beta = -1/\sqrt{5}$. Vậy:

$$F(n) = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^n$$

Phương trình hồi quy không đồng nhất
Tồn tại n để f(n) != 0

Đưa về dạng đồng nhất sau đó áp dụng lại cách làm như đã biết

Một số cách phân tích phương trình đệ quy phổ biến

Dạng 1:

$$\begin{aligned} T(n) &= T(n-1) + f(n) \\ &= T(0) + \sum_{j=1}^{n} f(j). \end{aligned}$$

Dạng 2: $T(n) = T(n/b) + f(n)$

$$\begin{aligned} T\left(b^k\right) &= T\left(b^{k-1}\right) + f\left(b^k\right) \\ &= T\left(b^{k-2}\right) + f\left(b^{k-1}\right) + f\left(b^k\right) \\ &= T(1) + \sum_{j=1}^{k} f\left(b^j\right) \\ &= T(1) + b\frac{b^k - 1}{b - 1}. \end{aligned}$$

Một số cách phân tích phương trình đệ quy phổ biến

Dạng 3: $T(n) = aT(n/b) + f(n$

$$T(n) = n^{\log_b a}\left[T(1) + \sum_{j=1}^{\log_b n} f\left(b^j\right)/a^j\right].$$

Dạng 4: MasterTheorem

$$T(n) = aT(n/b) + f(n) \quad \text{với } n = b^k, k = 1, 2, \ldots$$
$$T(1) = c,$$

với $a \geq 1, b \geq 2, c > 0$. Nếu $f(n) \in \Theta\left(n^d\right)$ với $d \geq 0$, thì

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

Bài tập trên lớp

Bài 1: $x(n) = x(n-1) + n \quad$ với $n > 0, \quad x(0) = 0$

Bài 2: $x(n) = x(n/3) + 1 \quad$ với $n > 1, \quad x(1) = 1 \quad (n = 3^k)$

Bài 3:

Đánh giá độ phức tạp của hàm f được cho bên dưới:

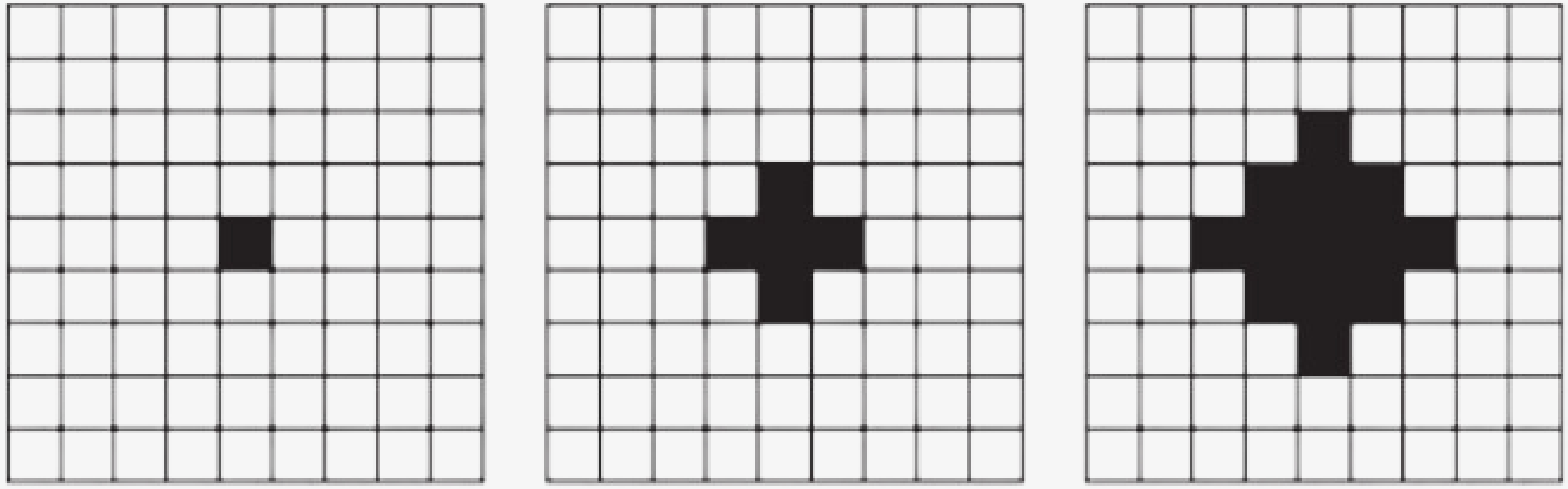
```
int f (int n)
{
        if (n==1)       return 2;
        return  3^{f(n/2)} + 2*log(f(n/2)) – f(n/2) +1;
}
```

Yêu cầu: Thành lập phương trình đệ quy (kèm giải thích ngắn gọn cách thành lập) và giải phương trình dùng **phương pháp truy hồi (còn gọi là thay thế).**

Bài tập trên lớp

Bài 4:

Thiết lập và giải hệ thứ truy hồi cho bài toán Von Neumann sẻ tìm số ô lân cận torng n bước.



$n = 0$ $n = 1$ $n = 2$

Bài tập về nhà

Bài 1:

Bài toán những con thỏ của Fibonacci. Một người đàn ông đặt 1 cặp thỏ vào 1 ô được bao bởi 4 bức tường. Hỏi có bao nhiêu cặp thỏ ở đó sau 1 năm nếu cặp thỏ (đực và cái) được bỏ vào ban đầu là mới sinh, và tất cả các cặp thỏ đều không có khả năng sinh sản trong tháng đầu tiên nhưng sau đó sẽ sinh ra một cặp đực/cái mới vào cuối mỗi tháng?

Bài tập về nhà

Bài 2:

Xét thuật toán đệ quy sau:

Algorithm Riddle ($A[0..n-1]$)

//Input: An array $A[0..n-1]$ of real numbers

if $n = 1$ return $A[0]$

else temp $\leftarrow$ Riddle($A[0..n-2]$)

    if temp $\leq A[n-1]$ return temp

    else return $A[n-1]$

a. Thuật toán này cho ra output là gì?

b. Thiết lập và giải hệ thức truy hồi của số lần thực hiện toán tử cơ bản của thuật toán.

THANK YOU