
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

BÀI TẬP MÔN PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN

HOMEWORK #02: Phân tích thuật toán đệ quy - Phần 1

GV hướng dẫn: Huỳnh Thị Thanh Thương

Nhóm thực hiện: Nhóm 5

1. Trần Văn Lực 20521587
2. Dương Thành Bảo Khanh 20521444
3. Lê Nguyễn Bảo Hân 20520174
4. Nguyễn Trần Minh Anh 20520394

TP.HCM, ngày 22 tháng 3 năm 2022

Mục Lục

Bài tập 1: Thành lập phương trình đệ quy

a). Gửi ngân hàng 1000 USD, lãi suất 12%/năm. Số tiền có được sau 30 năm là bao nhiêu?

```
long LaiXuat(int n)
{
   if (n == 0)
      return 1000;
   return 1.12 * (LaiXuat(n - 1));
}
```

Với bài toán lãi xuất thì ta có:

- số tiền có được sau n năm = tiền gốc *$(1 + lãi\ suất)^n$

  = tiền gốc *$(1 + lãi\ suất)^{n-1}$*$(1 + lãi\ suất)$

  = (Tiền có được sau n-1 năm) *$(1 + lãi\ suất)$

- Với F(n) là số tiền có được sau n năm :

  ➔ F(n) =F(n-1)*($1 + lãi\ suất$) = F(n-1)*1.12

Vậy ta có phương trình đệ quy

$$T(n) = \begin{cases} C_1 & \text{nếu } n = 0 \\ T(n-1) + C_2 & \text{nếu } n > 0 \end{cases}$$

❖ b).

```
long Fibo(int n)
{
   if (n == 0 || n == 1)
      return 1;
   return Fibo(n–
1)+Fibo(n–2);
}
```

Đối với bài toán tìm số Fibonacci, ta biết được giá trị của mỗi vị trí n là tổng giá trị của 2 giá trị ở ngay trước vị trí đang xét.

➔ Nên đối với 2 giá trị đầu tiên ở vị trí 0 và 1, giá trị của chúng luôn là 1.

➔ Còn với các vị trí từ 2 tăng dần, giá trị T(n) đang xét sẽ bằng tổng T(n – 1) và T(n – 2)

Từ các dữ kiện trên, ta thành lập được phương trình đệ quy: C_2

$$T(n) = \begin{cases} C_1 \text{ nếu } 0 \le n \le 1 \\ T(n-1) + T(n-2) + C_2 \text{ nếu } n > 1 \end{cases}$$

❖ c).

```
public int g(int n) {
   if (n == 1)
      return 2;
   else
      return 3 * g(n / 2) + g( n / 2) + 5;
}
```

- Gọi $T(n)$ là thời gian thực hiện $g(n)$
- Khi $n = 1$, chương trình thực hiện việc duy nhất là return 2. Ta có $T(1) = C_1$
- Khi $n > 1$, chương trình phải gọi đệ quy 2 lần $g(n/2)$, tốn thời gian là $2T(\frac{n}{2})$, sau đó cộng tổng và trả về kết quả. Thời gian thực hiện các phép toán và trả về kết quả là một hằng số C_2.
- Ta có phương trình đệ quy:

$$T(n) = \begin{cases} C_1, & khi\ n = 1 \\ 2T\left(\frac{n}{2}\right) + C_2, & khi\ n > 1 \end{cases}$$

❖ **d).**

```
long xn(int n)
{
  if (n == 0) return 1;
  long s = 0;
  for (int i=1; i<=n; i++)
    s = s + i*i*xn(n–i);
  return s;
}
```

- Gọi T(n) là thời gian thực hiện *xn(n)*.
- Khi n = 0, chương trình thực hiện duy nhất việc *return 1*. Ta có T(0) = C_1
- Khi n > 0, chương trình phải gọi đệ quy *xn(n-i)* n lần, nhân kết quả với với (i*i) và cộng tổng và return (với i chạy từ 1 tới n, bước tăng là 1). Thời gian thực hiện các phép toán và return là một hằng C_2.
  Ta có phương trình đệ quy:

$$T(n) = \begin{cases} C_1 & n = 0 \\ nT(n-1) + nC_2 & n > 0 \end{cases}$$

❖ **e).**

```
waste (n)
{    if (n==0) return 0;
     for (i = 1 to n )
          for (j = 1 to i )
               print i,j,n;
     for (i = 1 to 3)
               waste (n/2);
}
```

- Với T(n) là thời gian thực hiện *waste(n)*, ta có T(n/2) là thời gian thực hiện *waste(n/2)*.
- Khi n = 0, chương trình thực hiện duy nhất việc *return 0*. Ta có T(0) = C_1
- Khi n > 0, chương trình thực hiện gọi *waste(n/2)* ba lần, tốn thời gian 3T(n/2). Ngoài ra, thời gian để thực hiện *print i,j,n* là một hằng C_2.
  Ta có phương trình đệ quy:

$$T(n) = \begin{cases} C_1 & n = 0 \\ 3T(n/2) + n^2C_2 & n > 0 \end{cases}$$

❖ f).

```
Draw (n)
{    if (n < 1) return 0;
     for (i = 1 ; i <= n; i++ )
          for (j = 1 ; j <=n ; j++)
               print ("*");
     Draw (n-3);
}
```

- Gọi $T(n)$ là thời gian thực hiện $Draw(n)$
- Khi $n < 1$, chương trình thực hiện việc duy nhất là trả về kết quả là 0. Ta có $T(n) = C_1 \; khi \; n < 1$.
- Khi $n \geq 1$, chương trình thực hiện vòng lặp for ngoài n lần, vòng lặp for trong n^2 lần, thời gian thực hiện 2 vòng lặp là n^2C_2, sau đó thực hiện lệnh print ("*") tốn thời gian là một hằng C_3. Thời gian thực hiện 2 vòng lặp và lệnh print ("*") là $n^2C_2 + C_3$.
- Sau đó chương trình thực hiện đệ quy $Draw(n-3)$ tốn thời gian là $T(n-3)$.

- Ta có phương trình đệ quy:

$$T(n) = \begin{cases} C_1, & khi\ n < 1 \\ T(n-3) + n^2 C_2 + C_3, & khi\ n \geq 1 \end{cases}$$

g)

Gọi T(n) là số phép cộng cần thực hiện khi gọi Zeta (k). Hãy thiết lập công thức truy hồi cho T(n)

```
Cho hàm:
Zeta (n)
{   if (n == 0) Zeta = 6;
    else
    {   k = 0;
        Ret = 0;
        while (k<=n-1)
        {       Ret = Ret +Zeta(k) ;
                k = k+1;
        }
        Zeta = Ret;
}
```

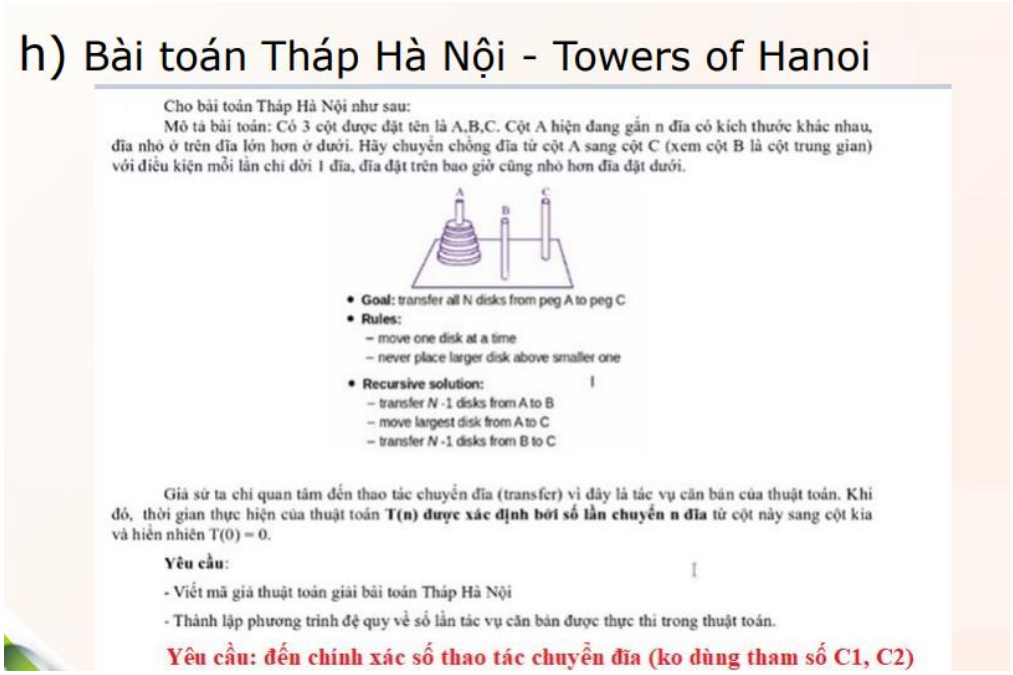
Dựa trên chương trình cho sẵn, ta có thể rút ra các dữ kiện sau:

➔ Tại vị trí 0, giá trị của hàm Zeta(0) bằng 6

➔ Đối với các vị trí từ 1 trở đi, hàm sẽ áp dụng một vòng lặp tính tổng các giá trị chạy từ 0 cho đến vị trí ngay trước nó: $T(n) = \sum_{k=0}^{n-1} T(k)$ + nC_3

Từ đó, ta suy ra được phương trình đệ quy của chương trình là:

$$T(n) = \begin{cases} C_1\ nếu\ n = 0 \\ T(n-1) + C_2\ nếu\ n > 0 \end{cases}$$



Theo yêu cầu đề và gợi ý giải pháp đệ quy ta có mã giả cho bài toán Tháp Hà Nội :

```
void Tower(n,A,B,C)
 {
        if(n==1)
      cout<<A<<"==>"<<C<<"\n";nếu n = 1 thì dịch chuyển từ A -> C
    else {
      Tower (n - 1, A, C, B);1. Dịch chuyển n-1 đĩa từ A -> B
      cout<<A<<"==>"<<C<<"\n";2. Dịch chuyển đĩa thứ n từ A -> C
      Tower (n - 1, B, A, C);3. Dịch chuyển n-1 đĩa từ B -> C
    }
 }
```

gọi T(n) = là số lần dịch chuyển n đĩa từ A -> C
Ta có bài toán trên được chia thành 3 bài toán :

1:Dịch chuyển n-1 đĩa từ A -> B T(n-1)

2: Dịch chuyển đĩa thứ n từ A -> C T(1) = 1

3: Dịch chuyển n-1 đĩa từ B -> C T(n-1)

⇨ T(n) = T(n-1) + T(n-1) +1 = $2T(n-1) + 1$

Vậy ta có phương trình đệ quy

$$T(n) = \begin{cases} 1 & \text{nếu } n = 1 \\ 2T(n-1) + 1 & \text{nếu } n > 1 \end{cases}$$

Bài tập 2: Giải các phương trình đệ quy sau bằng phương pháp truy hồi

1. T(n) = T(n-1) + 5 T(1) = 0

$$T(n) = T(n-1) + 5$$
$$= [T(n-2) + 5] + 5 = T(n-2) + 10$$
$$= [T(n-3) + 5] + 10 = T(n-3) + 15$$
$$= \cdots$$
$$T(n) = T(n-i) + 5i$$

Quá trình kết thúc khi $n - i = 1 \rightarrow i = n - 1$
$\rightarrow T(n) = T(1) + 5(n-1) = 5(n-1)$

2. $T(n) = T(n-1) + n$, $T(1) = 1$

Ta có:

$$T(n) = T(n-1) + n$$
$$= T(n-2) + (n-1) + n$$
$$= T(n-3) + (n-2) + (n-1) + n$$
$$= \cdots$$
$$= T(n-i) + (n-i+1) + (n-i+2) + \cdots + (n-1) + n$$
$$= T(n-i) + \sum_{k=0}^{i-1}(n-k)$$

Quá trình kết thúc khi $n - i = 1 \Leftrightarrow i = n - 1$

$$\Rightarrow T(n) = T(1) + \sum_{k=0}^{n-2}(n-k) = 1 + \sum_{k=0}^{n-2} n - \sum_{k=0}^{n-2} k$$
$$= 1 + n(n-1) - \frac{(n-1)(n-2)}{2} = \frac{n^2}{2} + \frac{n}{2}$$

3. T(n) = 3.T(n -1) + 1 T(1) = 4

= 3.[3.T(n – 2) + 1] + 1

= 9.T(n – 2) + 1 + 3

= 9.[3.T(n – 3) + 1] + 1 + 3

= 27.T(n – 3) + 1 + 3 + 9

= $3^i \cdot T(n-i) + \sum_{k=0}^{i-1} 3^k$

Quá trình chỉ kết thúc khi n – i = 1 ⇔ i = n – 1

⇒ T(n) = $3^{n-1} \cdot T(1) + \sum_{k=0}^{n-2} 3^k$

= $3^{n-1} \cdot 4 + \frac{3^{n-1}-1}{3-1}$

= $4.3^{n-1} + \frac{3^{n-1}-1}{2}$

4. T(n) = $\begin{cases} \mathbf{1} & \textbf{nếu n } = \mathbf{1} \\ \mathbf{2T(\frac{n}{2}) + 1} & \textbf{nếu n } > \mathbf{1} \end{cases}$

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

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

= 4[2T($\frac{n}{8}$) + 1] + 2 = 8T($\frac{n}{8}$) + (1 + 2 + 4)

Tại bước thứ i: T(n) = 2^i T($\frac{n}{2^i}$) + $2^i - 1$

Quá trình trên kết thúc khi $\frac{n}{2^i}$ = 1 => 2^i = n => i = $\log_2 n$

⇨ T(n) = n*T(1) + n-1

= 2n - 1

5. T(n) = $\begin{cases} \mathbf{1} & \textbf{nếu n } = \mathbf{1} \\ \mathbf{2T(\frac{n}{2}) + n} & \textbf{nếu n } > \mathbf{1} \end{cases}$

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

= 2[2T($\frac{n}{4}$) + $\frac{n}{2}$] + n = 4T($\frac{n}{4}$) + 2n

= 4[2T($\frac{n}{8}$) + $\frac{n}{4}$] + 2n = 8T($\frac{n}{8}$) + 3n

Tại bước thứ i: T(n) = 2^i T($\frac{n}{2^i}$) + i*n

Quá trình trên kết thúc khi $\frac{n}{2^i}$ = 1 => 2^i = n => i = $\log_2 n$

⇨ T(n) = n*T(1) + n$\log_2$ n

= n + n$\log_2$ n

6. $\boldsymbol{T(n) = 2T\left(\frac{n}{2}\right) + n^2}$ **T(1) = 1**

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

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

$$= 4 \cdot \left[2 \cdot T\left(\frac{n}{8}\right) + (\frac{n}{4})^2\right] + \frac{n^2}{2} + n^2$$

$$= 8T\left(\frac{n}{8}\right) + \frac{n^2}{4} + \frac{n^2}{2} + n^2$$

$$= 2^i \cdot T\left(\frac{n}{2^i}\right) + n^2 \sum_{k=0}^{i-1} \frac{1}{2}^k$$

Ta có: $\frac{n}{2^i} = 1 \Leftrightarrow i = log_2 n$

$$\Rightarrow T(n) = 2^{log_2 n} \cdot T(1) + n^2.(\frac{\frac{1}{2}^{log_2 n} - 1}{- 0.5})$$

$$= n - 2n^2.(\frac{1}{2}^{log_2 n} - 1)$$

7. $\boldsymbol{T(n) = 2T\left(\frac{n}{2}\right) + \log n,\ T(1) = 1}$

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

$$= 2[2T\left(\frac{n}{4}\right) + \log\left(\frac{n}{2}\right)] + \log n = 4T\left(\frac{n}{4}\right) + 2\log\left(\frac{n}{2}\right) + \log n$$

$$= 8T\left(\frac{n}{8}\right) + \log n + 2\log\frac{n}{2} + 4\log\frac{n}{4}$$

$$= 2^i T\left(\frac{n}{2^i}\right) + \sum_{k=0}^{i-1}(2^k \log(\frac{n}{2^k}))$$

$$= 2^i T\left(\frac{n}{2^i}\right) + \log n \sum_{k=0}^{i-1} 2^k - \log 2 \sum_{k=0}^{i-1} k2^k$$

$$= 2^i T\left(\frac{n}{2^i}\right) + \log n \left(2^i - 1\right) - \log 2 \left[(i-2)2^i + 2\right]$$

$$T(n) = \begin{cases} 1, nếu\ n = 1 \\ 2^i T\left(\frac{n}{2^i}\right) + \log n \left(2^i - 1\right) - \log 2 \left[(i-2)2^i + 2\right], nếu\ n > 1 \end{cases}$$

Quá trình kết thúc khi $\frac{n}{2^i} = 1 \Leftrightarrow i = \log_2 n$

Khi đó:

$$T(n) = 2^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + \log n \left(2^{\log_2 n} - 1\right) - \log 2 \left[(\log_2 n - 2)2^{\log_2 n} + 2\right]$$

$$T(n) = n + (n-1)\log n - \log 2\,(n \log_2 n - 2n + 2)$$

$$T(n) = n + n \log n - \log n - n \log n + 2n + 2$$

$$T(n) = 3n - \log n - 2$$

8. T(n) = T(n-1) + T(n-2) T(0) = T(1) = 1

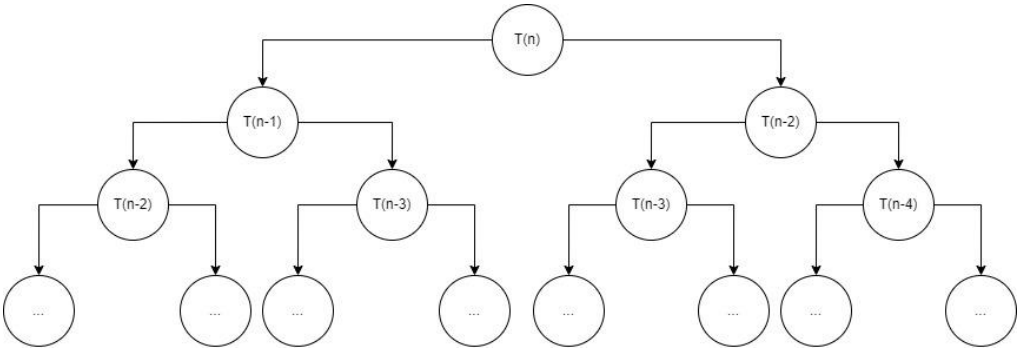
Giả sử $T(n-1) \approx T(n-2)$

- *Với cận dưới:*

  $T(n) = 2T(n-2)$
  $= 2[2T(n-4)] = 4T(n-4)$
  $= 4[2T(n-6) = 8T(n-6)$
  $= \cdots$
  $T(n) = 2^i * T(n-2i)$
  Quá trình kết thúc khi $n - 2i = 0 \rightarrow i = n/2$
  $\rightarrow T(n) = 2^{n/2} * T(0) = 2^{n/2}$

- *Với cận trên:*

  $T(n) = 2T(n-1)$
  $= 2[2T(n-2)] = 4T(n-2)$
  $= 4[2T(n-3) = 8T(n-3)$
  $= \cdots$
  $T(n) = 2^i * T(n-i)$
  Quá trình kết thúc khi $n - i = 0 \rightarrow i = n$
  $\rightarrow T(n) = 2^n * T(0) = 2^n$



Mỗi lần gọi đến F(n) sẽ phải thực hiện thêm 2 lần gọi đến F làm nhân đôi chi phí. Vậy, thời gian của chương trình tăng theo cấp số nhân hay T(n) ≈ 2^n. Tuy nhiên, có thể thấy tỉ lệ giữa các phần tử trong chuỗi tính toán gần với tỉ lệ vàng. Cận trên chặt chẽ nhất là T(n) ≈ φ^n với $\varphi = \frac{1+\sqrt{5}}{2}$

Bài tập 3: Giải phương trình đệ quy sau bằng phương pháp truy hồi với T(1) = 1

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

$$T(n) = 3\left[3T\left(\frac{n}{4}\right) + \left(\frac{n}{2}\right)^2\right] + n^2 = 9T\left(\frac{n}{4}\right) + \left(1 + \frac{3}{4}\right)n^2$$

$$T(n) = 9\left[3T\left(\frac{n}{8}\right) + \left(\frac{n}{4}\right)^2\right] + \left(1 + \frac{3}{4}\right)n^2 = 27T\left(\frac{n}{8}\right) + \left(1 + \frac{3}{4} + \frac{9}{16}\right)n^2$$

…

$$T(n) = 3^i T\left(\frac{n}{2^i}\right) + n^2 \sum_{k=0}^{i-1}\left(\frac{3}{4}\right)^k$$

Ta có:

$$T(n) = \begin{cases} 1, \ n = 1 \\ 3^i T\left(\frac{n}{2^i}\right) + n^2 \sum_{k=0}^{i-1}\left(\frac{3}{4}\right)^k, n > 1 \end{cases}$$

Quá trình kết thúc khi $\frac{n}{2^i} = 1 \Leftrightarrow i = \log_2 n$

Khi đó:

$$T(n) = 3^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + n^2 \sum_{k=0}^{\log_2 n - 1}\left(\frac{3}{4}\right)^k$$

$T(n) = 3^{\log_2 n} + \frac{n^2\left(1-\left(\frac{3}{4}\right)^{\log_2 n}\right)}{1-\frac{3}{4}} = n^{\log_2 3} + 4n^2 - 4n^{\log_2 3}$

$T(n) = 4n^2 - 3n^{\log_2 3}$

2. T(n) = $\begin{cases} \mathbf{1} & \textbf{nếu n } = \mathbf{1} \\ \mathbf{8\ T(\frac{n}{2})\ +\ n^3} & \textbf{nếu n } > \mathbf{1} \end{cases}$

T(n) $= 8T(\frac{n}{2}) + n^3$

$= 8[8T(\frac{n}{4}) + \frac{n^3}{8}\ ]\ + n^3 \quad = 64T(\frac{n}{4}) + 2n^3$

$= 64[8T(\frac{n}{8}) + \frac{n^3}{64}\ ] + 2n^3 \qquad = 512T(\frac{n}{8}) + 3n^3$

Tại bước thứ i: $T(n) = 2^{3i}\ T(\frac{n}{2^i})\ +\ i * n^3$

Quá trình trên kết thúc khi $\frac{n}{2^i} = 1 \Rightarrow 2^i = n \Rightarrow i = \log_2 n$

$\Rightarrow T(n) = n^3 * T(1) + \log_2 n * n^3$

$= n^3 + (\log_2 n) * n^3 = n^3(1 + log_2 n)$

3.

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

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

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

$= \cdots$

$$T(n) = 4^i * T\left(\frac{n}{3^i}\right) + \sum_{k=0}^{i-1}\left(\left(\frac{4}{3}\right)^k n\right)$$

Quá trình kết thúc khi $\frac{n}{3^i} = 1 \rightarrow i = log_3 n$

$$\rightarrow T(n) = 4^{log_3 n} * T(1) + n * \sum_{k=0}^{log_3 n - 1}\left(\frac{4}{3}\right)^k$$

$$= 4^{log_3 n} + n * \frac{1 - \left(\frac{4}{3}\right)^{log_3 n}}{1 - \frac{4}{3}}$$

4. $\boldsymbol{T(n) = 9T\left(\frac{n}{3}\right) + n^2}$

$$= 9 \cdot \left[9 \cdot T\left(\frac{n}{9}\right) + \left(\frac{n}{3}\right)^2\right] + n^2$$

$$= 81 \cdot T\left(\frac{n}{9}\right) + n^2 + n^2$$

$$= 81 \cdot \left[9 \cdot T\left(\frac{n}{27}\right) + \left(\frac{n}{9}\right)^2\right] + 2n^2$$

$$= 729 \cdot T\left(\frac{n}{27}\right) + n^2 + 2n^2$$

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

Ta có: $\frac{n}{3^i} = 1 \Leftrightarrow i = log_3 n$

$$\Rightarrow T(n) = 3^{2\,log_3 n} + log_3 n \cdot n^2$$

$$= n^2 + log_3 n \cdot n^2$$

$$= n^2(1 + log_3 n)$$

5. $\boldsymbol{T(n) = 2T\left(\sqrt{n}\right) + 1}$

Ta có:

$T(n) = 2T\left(\sqrt{n}\right) + 1$

$$= 2\left[2T\left(\sqrt{\sqrt{n}}\right) + 1\right] + 1 = 4T\left(n^{\frac{1}{4}}\right) + (1 + 2)$$

$$= 4\left[2T\left(n^{\frac{1}{8}}\right) + 1\right] + (1 + 2) = 8T\left(n^{\frac{1}{8}}\right) + (1 + 2 + 4)$$

…

$T(n) = 2^i T\left(n^{\frac{1}{2i}}\right) + \sum_{k=0}^{i-1} 2^k$

Ta có:

$$T(n) = \begin{cases} 0, khi\ n = 2 \\ 2^i T\left(n^{\frac{1}{2i}}\right) + \sum_{k=0}^{i-1} 2^k\ , khi\ n > 2 \end{cases}$$

Quá trình kết thúc khi $n^{\frac{1}{2i}} = 2 \Leftrightarrow i = \log_2(\log_2 n)$

Khi đó:

$$T(n) = 2^{\log_2(\log_2 n)} T\left(n^{\frac{1}{2^{\log_2(\log_2 n)}}}\right) + \sum_{k=0}^{\log_2(\log_2 n)-1} 2^k$$

$$= \log_2 n\, T(2) + \sum_{k=0}^{\log_2(\log_2 n)-1} 2^k$$

$$= \log_2 n * 0 + \log_2 n - 1$$

$$T(n) = \log_2 n - 1$$

Bài tập 4: Giải phương trình đệ quy sau dùng phương trình đặc trưng

a.T(n) = 4T(n-1) -3T(n - 2) và T(0) = 1 ,T(1) = 2

Ta có :

T(n) - 4T(n-1) + 3T(n - 2) = 0

Dặt T(n) = X^n đưa về dạng ẩn X :

⇨ $X^n - 4\,X^{n-1} + 3X^{n-2} = 0$

Phương trình đặc trưng : $X^2 - 4X + 3 = 0$

⇨ (X-3)(X-1) = 0

⇨ phương trình có 2 ngiệm đơn X=3 và X = 1

T(n) = $3^n C_1 + 1^n C_2$

Với T(0) = 1 => $C_1 + C_2 = 1$

Với T(1) = 2 => $3C_1 + C_2 = 2$

Giải hệ phương trình trên, ta dược : $C_1 = C_2 = \frac{1}{2}$

➔ Kết luận: T(n) = $\frac{1}{2} 3^n + \frac{1}{2}$

b) $T(n) = 10T(n-1) - 33T(n-2) + 36T(n-3)$ và

T(0) = 0, T(1) = 1, T(2) = 7

- Xét phương trình dạng:
  $$T(n) - 10T(n-1) + 33T(n-2) - 36T(n-3) = 0$$
- Đặt , đưa (1) về dạng phương trình ẩn X:

$$X^n - 10X^{n-1} + 33X^{n-2} - 36X^{n-3} = 0$$

$$\Leftrightarrow X^{n-3}(X^3 - 10X^2 + 33X - 36) = 0$$

$$\Leftrightarrow X^{n-3} = 0 \text{ hoặc } X^3 - 10X^2 + 33X - 36 = 0 \text{ (2)}$$

- Phương trình đặc trưng: $X^3 - 10X^2 + 33X - 36 = 0$ có 1 nghiệm đơn $X_1 = 4$ và một nghiệm kép $X_2 = 3$

$$T(n) = C_1 4^n + C_2 3^n + C_3 n 3^n$$

- Dựa vào T(0), T(1), T(2) để tìm tham số, ta có:
  - T(0) = 0 → $C_1 + C_2 = 0$
  - T(1) = 1 → $4C_1 + 3C_2 + 3C_3 = 1$
  - T(2) = 7 → $16C_1 + 9C_2 + 18C_3 = 7$

Giải hệ phương trình, ta được: $C_1 = 1$, $C_2 = -1$, $C_3 = 0$

⇨ Kết luận: $T(n) = 4^n - 3^n$

c) $T(n) = T(n-1) + T(n-2)$ $T(0) = 1,\ T(1) = 1$

- Xét chương trình dạng:
  $T(n) - T(n-1) - T(n-2) = 0$

- Đặt $T(n) = X^n$, đưa về dạng phương trình ẩn X:
  $X^n - X^{n-1} - X^{n-2} = 0$

- Ta có phương trình đặc trưng:
  $X^2 - X^x - 1 = 0$

  Phương trình có 2 nghiệm đơn $X = \frac{1 \pm \sqrt{5}}{2}$

  $T(n) = c_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + c_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$

Ta có: $\begin{cases} T(0) = 1 \\ T(1) = 1 \end{cases} \rightarrow \begin{cases} c_1 + c_2 = 1 \\ \frac{1+\sqrt{5}}{2} c_1 + \frac{1-\sqrt{5}}{2} c_2 = 1 \end{cases} \rightarrow \begin{cases} c_1 = \frac{5+\sqrt{5}}{10} \\ c_2 = \frac{5-\sqrt{5}}{10} \end{cases}$

Vậy, $T(n) = \frac{5+\sqrt{5}}{10} * \left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{5-\sqrt{5}}{10} * \left(\frac{1-\sqrt{5}}{2}\right)^n$