CANTHO UNIVERSITY

CHƯƠNG 1

KỸ THUẬT
PHÂN TÍCH THUẬT TOÁN

Tuần 3 – GIẢI PHƯƠNG TRÌNH ĐỆ QUY

Dạng phương trình đệ quy

Dạng tổng quát của một **phương trình đệ quy** là:

$$T(n) = \begin{cases} C(n) \\ F(T(k)) + d(n) \end{cases}$$

← *Khi đệ quy dừng*

← *Khi đệ quy chưa dừng*

- **C(n)**: thời gian thực hiện chương trình ứng với trường hợp đệ quy dừng.
- **F(T(k)):** hàm xác định thời gian theo T(k).
- **d(n)**: thời gian phân chia bài toán và tổng hợp các kết quả.

Giải phương trình đệ quy

- Có 3 phương pháp giải phương trình đệ quy:

(1) Phương pháp truy hồi.

- Triển khai T(n) theo T(n-1), rồi T(n-2), … cho đến T(1) hoặc T(0)
- Suy ra nghiệm

(2) Phương pháp đoán nghiệm.

- Dự đoán nghiệm f(n)
- Áp dụng định nghĩa tỷ suất tăng và chứng minh f(n) là tỷ suất tăng của T(n)

(3) Phương pháp lời giải tổng quát.

Giải phương trình đệ quy

- Có 3 phương pháp giải phương trình đệ quy:

(1) Phương pháp truy hồi.

- Triển khai T(n) theo T(n-1), rồi T(n-2), … cho đến T(1) hoặc T(0)
- Suy ra nghiệm

(2) Phương pháp đoán nghiệm.

- Dự đoán nghiệm f(n)
- Áp dụng định nghĩa tỷ suất tăng và chứng minh f(n) là tỷ suất tăng của T(n)

(3) Phương pháp lời giải tổng quát.

Phương pháp truy hồi

- Dùng đệ quy để thay thế T(m) với m < n (vào phía phải phương trình) cho đến khi tất cả T(m) với m>1 được thay thế bởi biểu thức của T(1) hoặc T(0).

Triển khai T(n) theo
T(n-1) rồi đến
T(n-2) tiếp đến
...
cho đến T(1)

- Vì T(1) và T(0) là hằng số nên công thức T(n) chứa các số hạng chỉ liên quan đến n và hằng số.
- Từ công thức đó suy ra nghiệm của phương trình.

Ví dụ 1. Giải PTĐQ bằng PP truy hồi (n!)

Hàm tính Giai thừa n!

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

Ta có : $T(n) = \mathbf{T(n-1)} + C_2$ **(1)**

$= [T((n-1)-1) + C_2] + C_2 = \mathbf{T(n-2)} + 2C_2$ **(2)**

$= [T((n-2)-1) + C_2] + 2C_2 = \mathbf{T(n-3)} + 3C_2$ **(3)**

Quy luật truy hồi

……

$T(n) = T(\mathbf{n-i}) + iC_2$ **(i)**

$\mathbf{= T(0) = C_1}$

- Quá trình truy hồi kết thúc khi **n - i = 0** hay **<u>i = n</u>**

$\Rightarrow T(n) = T(0) + nC_2 = C_1 + nC_2 = O(\mathbf{max}(1, n)) = \mathbf{O(n)}$

Ví dụ 2. Giải PTĐQ bằng PP truy hồi (MergeSort)

CANTHO UNIVERSITY

Hàm MergeSort

$$T(n) = \begin{cases} C_1 & \text{nếu } n=1 \\ 2T(\frac{n}{2}) + nC_2 & \text{nếu } n>1 \end{cases}$$

$T(n/2)=2T((n/2)/2)+(n/2)C_2$

- $T(n) = 2T(n/2) + nC_2$ **(1)**

$= 2[2T(n/4) + (n/2)C_2] + nC_2 = 4T(n/4) + 2nC_2$ **(2)**

$= 4[2T(n/8) + (n/4)C_2] + 2nC_2 = 8T(n/8) + 3nC_2$ **(3)**

Quy luật truy hồi

$= \ldots\ldots\ldots$

$= \boxed{2^i T(n/2^i) + inC_2}$ **(i)**

$\boxed{= T(n/2^i) = T(1) = C_1}$

Quá trình truy hồi kết thúc khi $n/2^i = 1$ hay $2^i = n \rightarrow$ <u>$i = \log n$</u>

$\Rightarrow T(n) = nT(1) + \log n.nC_2 = nC_1 + n\log nC_2 = O(\max(n, n\log n)) = O(n\log n)$

Ví dụ 3. Giải PTĐQ bằng PP truy hồi (Tháp Hà nội)

Tháp Hà nội

$$T(n) = \begin{cases} C1 & n = 1 \\ 2T(n-1) + C2 & n > 1 \end{cases}$$

$$T(n) = 2\,T(n-1) + C_2$$

$$= 2\,[2T(n-2) + C_2] + C_2 = 4T(n-2) + 3C_2$$

$$= 4\,[2T(n-3) + C_2] + 3C_2 = 8T(n-3) + 7C_2$$

....

$$= 2^i\,T(n - i) + (2^i - 1)C_2$$

Quá trình truy hồi dừng khi **$n - i = 1$** hay $i = n - 1$.

$$\Rightarrow T(n) = 2^{n-1}\,T(1) + (2^{n-1} - 1)C_2$$

$$= 2^{n-1}\,C_1 + (2^{n-1} - 1)C_2 = O(\max(2^n, 2^n) = O(2^n)$$

Bài tập

Giải các phương trình đệ quy sau với **T(1) = 1** và:

1. $T(n) = T(n/2) + 1$
2. $T(n) = 3T(n/2) + n$
3. $T(n) = 4T(n/2) + n^2$

Bài giải (1)

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

$$
\begin{aligned}
T(n) &= \mathbf{T(n/2)} + 1 \\
&= \mathbf{[T(n/2^2) + 1]} + 1 = \mathbf{T(n/2^2)} + 2 \\
&= \mathbf{[T(n/ 2^3) + 1]} + 2 = \mathbf{T(n/2^3)} + 3 \\
&\ldots\ldots\ldots \\
&= T(\mathbf{n/2^i}) + i
\end{aligned}
$$

Quá trình truy hồi dừng lại khi $\mathbf{n/2^i = 1}$ hay $2^i = n \rightarrow i = \log n$.

$\Rightarrow T(n) = T(1) + \log n = 1 + \log n = O(\max(1, \log n)) = \mathbf{O(\log n)}$

Bài giải (2)

- **$T(1) = 1, T(n) = 3T(n/2)+n$**
- $T(n) = 3\mathbf{T(n/2)} + n$

$$= 3\ [\mathbf{3T(n/2^2) + n/2}] + n = 3^2\ \mathbf{T(n/2^2)} + 5n/2$$

$$= 3^2\ [\mathbf{3T(n/2^3) + n/2^2}] + 5n/2 = 3^3\ \mathbf{T(n/2^3)} + 19n/2^2$$

....

$$= 3^i\ T(n/\ 2^i) + (3^i - 2^i)n/2^{i-1}$$

Quá trình truy hồi dừng lại khi $n/2^i = 1$ hay $2^i = n$ và $i=\log n$.

$$\Rightarrow T(n) = 3^{\log n}\ T(1) + (3^{\log n} - 2^{\log n})\ n/2^{\log n-1}$$

$$= 3^{\log n} + (3^{\log n} - 2^{\log n})\ n/2^{\log n-1} = \mathbf{O(3^{\log n})}$$

Bài giải (3)

- **$T(1) = 1, T(n) = 4T(n/2) + n^2$**

- $T(n) = 4\mathbf{T(n/2)} + n^2$

$= 4\ [\mathbf{4T(n/2^2) + (n/2)^2}] + n^2 = 4^2\ \mathbf{T(n/2^2)} + 2n^2$

$= 4^2\ [\mathbf{4T(n/2^3) + (n/2^2)^2}] + 2n^2 = 4^3\ \mathbf{T(n/2^3)} + 3n^2$

....

$= 4^i\ T(\mathbf{n/2^i}) + in^2$

Quá trình truy hồi dừng lại khi $\mathbf{n/2^i = 1}$ hay $2^i = n$ và $i = \log n$.

$\Rightarrow T(n) = 4^i\ T(n/\ 2^i) + in^2 = 4^{\log n} + \log n.n^2 = 2^{2\ \log n} + n^2 \log n$

$= 2^{\log n^2} + n^2 \log n = n^2 + n^2 \log n = \mathbf{O(n^2 \log n)}$

Một số công thức Logarit

CANTHO UNIVERSITY

10. Logarit : $0 < N_1, N_2, N$ và $0 < a, b \neq 1$ ta có

$\log_a N = M \Leftrightarrow N = a^M$	$\log_a\left(\frac{N_1}{N_2}\right) = \log_a N_1 - \log_a N_2$
$\log_a a^M = M$	$\log_a N^\alpha = \alpha \log_a N$
$a^{\log_a N} = N$	$\log_{a^\alpha} N = \frac{1}{\alpha}\log_a N$
$N_1^{\log_a N_2} = N_2^{\log_a N_1}$	$\log_a N = \frac{\log_b N}{\log_b a}$
$\log_a(N_1.N_2) = \log_a N_1 + \log_a N_2$	$\log_a b = \frac{1}{\log_b a}$

Phương pháp Lời giải tổng quát

- Trong mục này, ta sẽ nghiên cứu các phần sau:
  - Bài toán đệ quy tổng quát.
  - Thành lập phương trình đệ quy tổng quát.
  - Giải phương trình đệ quy tổng quát.
  - Các khái niệm về *nghiệm thuần nhất, nghiệm riêng* và *hàm nhân*.
  - Nghiệm của phương trình đệ quy tổng quát khi **d(n) là hàm nhân.**
  - Nghiệm của phương trình đệ quy tổng quát khi **d(n) không phải là hàm nhân.**

Bài toán đệ quy tổng quát

- **Giải thuật Chia để trị**:

  - Phân rã bài toán kích thước n thành a bài toán con có kích thước n/b.

  - Giải các bài toán con và tổng hợp kết quả.

- Với các bài toán con, tiếp tục *Chia để trị* đến khi các bài toán con kích thước 1 → *Thuật toán đệ quy*.
- **Giả thiết :**

  - Bài toán con kích thước 1 lấy **một** đơn vị thời gian

  - Thời gian chia bài toán và tổng hợp kết quả để được lời giải của bài toán ban đầu là **d(n)**.

Thành lập phương trình đệ quy tổng quát

- Gọi T(n) = thời gian giải bài toán kích thước n,

  T(n/b) = thời gian giải bài toán con kích thước n/b.
- *Khi n = 1*: T(1) = **1**.
- *Khi n > 1*: giải **a bài toán con kích thước n/b** tốn **aT(n/b)** + thời gian phân chia bài toán và tổng hợp là **d(n)**.
- Vậy phương trình **đệ quy tổng quát** có dạng:

$$T(1) = 1$$
$$T(n) = aT(n/b) + d(n)$$

a: *số lượng bài toán con*

(n/b): *kích thước bài toán con*

Ví dụ MergeSort

```
FUNCTION MergeSort (L:List; n:Integer):List;
VAR  L1,L2:List;
BEGIN
  IF n=1 THEN RETURN(L)
  ELSE BEGIN
     Chia đôi L thành L1 và L2, với độ dài n/2;
     RETURN(Merge(MergeSort(L1,n/2),MergeSort(L2,n/2)));
   END;
END;
```

- *Khi n = 1*: tốn C_1
- *Khi n > 1*: a = b= 2 (Giải đệ quy 2 bài toán con kích thước n/2, nên cần 2T(n/2)).
- Thời gian tổng hợp kết quả (hàm Merge: thời gian tăng theo n) : d(n) = nC_2.
- Vậy phương trình đệ quy:

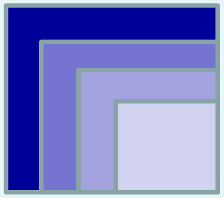
$$T(n) = \begin{cases} C_1 & \text{nếu n=1} \\ 2T(\frac{n}{2}) + nC_2 & \text{nếu n>1} \end{cases}$$

Phương trình đệ quy

Chương trình ĐỆ QUY

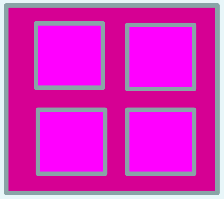
ĐỆ QUY **Lùi**

→ *PP truy hồi*



ĐỆ QUY **Chia để trị**

→ *PP truy hồi/**PP lời giải TQ***



Phương trình đệ quy:

$$T(n) = \begin{cases} C(n) \\ F(T(k)) + d(n) \end{cases}$$

Phương trình đệ quy **tổng quát:**

$$T(1) = 1$$
$$T(n) = aT(n/b) + d(n)$$

CANTHO UNIVERSITY

Ví dụ các dạng phương trình đệ quy

ĐỆ QUY **Lùi**

Hàm tính Giai thừa n!

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

Tháp Hà nội

$$\mathrm{T(n)} = \begin{cases} \mathrm{C1} & \mathrm{n} = 1 \\ \mathrm{2T(n-1)+C2} & \mathrm{n} > 1 \end{cases}$$

ĐỆ QUY **Chia để trị**

Phương trình đệ quy ***tổng quát***

Hàm MergeSort

$$T(n) = \begin{cases} C_1 & \text{nếu n=1} \\ 2T(\frac{n}{2}) + nC_2 & \text{nếu n>1} \end{cases}$$

1. **T(n) = T(n/2) + 1**
2. **T(n) = 3T(n/2) + n**
3. **T(n) = 4T(n/2) + n^2**

CANTHO UNIVERSITY

Giải phương trình đệ quy tổng quát

$$T(n)=\begin{cases}1 & \text{neu } n=1\\ aT\left(\frac{n}{b}\right)+d(n) & \text{neu } n>1\end{cases}$$

Dùng phương pháp truy hồi với giả thiết ***$n=b^k$***

$$T(n)=\boxed{aT\left(\frac{n}{b}\right)}+\underbrace{d(n)}$$

$$T(n)=a\left[aT\left(\frac{n}{b^2}\right)+d\left(\frac{n}{b}\right)\right]+d(n) \quad =\boxed{a^2T\left(\frac{n}{b^2}\right)}+\underbrace{ad\left(\frac{n}{b}\right)+d(n)}$$

$$T(n)=a^2\left[aT\left(\frac{n}{b^3}\right)+d\left(\frac{n}{b^2}\right)\right]+ad\left(\frac{n}{b}\right)+d(n) \quad =\boxed{a^3T\left(\frac{n}{b^3}\right)}+\underbrace{a^2d\left(\frac{n}{b^2}\right)+ad\left(\frac{n}{b}\right)+d(n)}$$

....

$$T(n)=\boxed{a^iT\left(\frac{n}{b^i}\right)}+\underbrace{\sum_{j=0}^{i-1}a^jd\left(\frac{n}{b^j}\right)}$$

Giải phương trình đệ quy tổng quát

$$T(n) = a^i T\left(\frac{n}{b^i}\right) + \sum_{j=0}^{i-1} a^j d\left(\frac{n}{b^j}\right)$$

Giả sử $n = b^k$, quá trình suy rộng trên sẽ kết thúc khi i = k.
Khi đó ta được:

$$T\left(\frac{n}{b^i}\right) = T\left(\frac{n}{b^k}\right) = T\left(\frac{b^k}{b^k}\right) = T(1) = 1$$

Thay vào trên ta có:

$$T(n) = a^k + \sum_{j=0}^{k-1} a^j d\left(b^{k-j}\right)$$

Nghiệm thuần nhất và nghiệm riêng

$$T(n) = \underbrace{a^{k}}_{\text{Nghiệm thuần nhất}} + \underbrace{\sum_{j=0}^{k-1} a^{j} d\left(b^{k-j}\right)}_{\text{Nghiệm riêng}}$$

Nghiệm thuần nhất là nghiệm chính xác khi d(n) = 0, ∀n : biểu diễn thời gian giải tất cả bài toán con.

Nghiệm riêng phụ thuộc hàm tiến triển d(n), số lượng (a) và kích thước bài toán con (n/b): biểu diễn thời gian tạo bài toán con và tổng hợp kết quả.

Nghiệm thuần nhất và nghiệm riêng

$$T(n) = \boxed{a^k} + \boxed{\sum_{j=0}^{k-1} a^j d\left(b^{k-j}\right)}$$

Nghiệm thuần nhất (NTN)
$a^k = n^{\log_b a}$

Nghiệm riêng (NR)

NTN: Do $n=b^k$ nên $n^{\log_b a} = (b^k)^{\log_b a} = b^{\log_b a^k} = a^k$

Nghiệm của phương trình là: MAX(NTN,NR).

Hàm nhân

- Hàm f(n) là **hàm nhân** (multiplicative function) nếu:

  f(m.n) = f(m).f(n) $\forall$m, n nguyên dương

- **Ví dụ:**
  - Hàm **f(n) = 1** là hàm nhân vì: **f(m.n) =1.1=1=f(m).f(n)**
  - Hàm **f(n) = n** là hàm nhân vì: **f(m.n)=m.n=f(m).f(n)**
  - Hàm **f(n) = n^k** là hàm nhân vì: **f(m.n) = $(m.n)^k = m^k.n^k$ = f(m).f(n)**

    Chẳng hạn k=2: $(2.3)^2 = 6^2 = 36 = 4.9 = 2^2.3^2$
  - Hàm **f(n) = logn** *không phải* là hàm nhân vì:

  f(m.n) = log(m.n) = logm+logn ≠ logm.logn = f(m).f(n)

Ghi nhớ: Các hàm độ phức tạp thường gặp đều là hàm nhân, trừ hàm logarit

Tính nghiệm riêng khi d(n) là hàm nhân

- Khi d(n) là **hàm nhân**, ta có:

$d(b^{k-j}) = \mathbf{d(b.b.b\ldots b)} = \mathbf{d(b).d(b)\ldots d(b)} = [d(b)]^{k-j}$

Hàm nhân f(m.n) = f(m).f(n)

$$NR = \sum_{j=0}^{k-1} a^j d\left(b^{k-j}\right) = \sum_{j=0}^{k-1} a^j [d(b)]^{k-j} = [d(b)]^k \sum_{j=0}^{k-1} \left[\frac{a}{d(b)}\right]^j = [d(b)]^k \frac{\left[\frac{a}{d(b)}\right]^k - 1}{\frac{a}{d(b)} - 1}$$

$$\text{Hay } \boxed{NR = \frac{a^k - [d(b)]^k}{\frac{a}{d(b)} - 1}}$$

Một số tổng thông dụng

$S = 1 + 2 + 3 + \ldots + n = n(n+1)/2 \approx n^2/2$

$S = 1 + 2^2 + 3^2 + \ldots + n^2 = n(n+1)(2n+1)/6 \approx n^3/3$

$S = 1 + a + a^2 + a^3 + \ldots + a^n = (a^{n+1} - 1)/(a-1)$

Nếu $0 < a < 1$ thì

$S \leq 1/(1-a)$

và khi $n \to \infty$ thì

S tiến về $1/(1-a)$

$S = 1 + 1/2 + 1/3 + \ldots + 1/n = \ln(n) + \gamma$

Hằng số Euler $\gamma \approx 0.577215665$

$S = 1 + 1/2 + 1/4 + 1/8 + \ldots + 1/2^n + \ldots \approx 2$

Ba trường hợp

$$NR = \frac{a^k - [d(b)]^k}{\frac{a}{d(b)} - 1}$$

- **Trường hợp 1**: **a > d(b)**

  Trong công thức trên: $a^k > [d(b)]^k$

  Theo quy tắc tính độ phức tạp:

  NR là $O(a^k) = O(n^{\log_b a}) = NTN$.

  Do đó $\boxed{T(n) = O(n^{\log_b a})}$

Nhận xét : T(n) chỉ phụ thuộc vào a, b mà không phụ thuộc hàm tiến triển d(n) → Cải tiến thuật toán = **Giảm a** : giảm số bài toán con.

Ba trường hợp

$$NR = \frac{a^k - [d(b)]^k}{\frac{a}{d(b)} - 1}$$

- **Trường hợp 2**: **a < d(b)**

  Trong công thức trên: $[d(b)]^k > a^k$

  Theo quy tắc tính độ phức tạp :

  NR là $O([d(b)]^k) = O(n^{\log_b d(b)}) >$ NTN.

  Do đó $\boxed{T(n) = O(n^{\log_b d(b)})}$

Nhận xét : T(n) phụ thuộc vào b và hàm tiến triển d(b) → Cải tiến thuật toán = **Giảm d(b)** : cải tiến việc phân chia bài toán và tổng hợp kết quả.

Ba trường hợp

$$NR = \frac{a^k - [d(b)]^k}{\frac{a}{d(b)} - 1}$$

Trường hợp 3: **a = d(b)**

Công thức trên không xác định nên phải tính trực tiếp nghiệm riêng:

$$NR = [d(b)]^k \sum_{j=0}^{k-1} \left[\frac{a}{d(b)}\right]^j = a^k \sum_{j=0}^{k-1} 1 = a^k k \qquad (\text{do } a = d(b))$$

Do $n = b^k$ nên $k = \log_b n$ và $a^k = n^{\log_b a}$.

Vậy NR là $n^{\log_b a}\log_b n > NTN$.

Do đó $\boxed{T(n) = O(n^{\log_b a}\log_b n)}$

3 trường hợp của NR khi d(n) là hàm nhân

Trường hợp 1: $a > d(b) \rightarrow T(n) = O(n^{\log_b a})$

Trường hợp 2: $a < d(b) \rightarrow T(n) = O(n^{\log_b d(b)})$

Trường hợp 3: $a = d(b) \rightarrow T(n) = O(n^{\log_b a} \log_b n)$

Nghiệm của phương trình đệ quy tổng quát

$$T(n) = a^k + \sum_{j=0}^{k-1} a^j d\left(b^{k-j}\right)$$

Nghiệm thuần nhất (NTN)

Nghiệm riêng (NR)

NTN = $O(a^k) = O(n^{\log_b a})$

NR *khi d(n) là hàm nhân*

- $(1)\ a > d(b) \rightarrow T(n) = O(n^{\log_b a})$
- $(2)\ a < d(b) \rightarrow T(n) = O(n^{\log_b d(b)})$
- $(3)\ a = d(b) \rightarrow T(n) = O(n^{\log_b a} \log_b n)$

Nghiệm của phương trình T(n) = MAX(**NTN, NR**)

CANTHO UNIVERSITY

Tính nghiệm riêng khi d(n) không phải là hàm nhân

- Trường hợp *hàm tiến triển d(n)* ***<u>không phải</u>*** *là hàm nhân:* không thể áp dụng các công thức ứng với 3 trường hợp nói trên mà phải **tính trực tiếp** NR, sau đó so sánh với NTN và lấy **MAX(NR,NTN).**

Quy tắc chung để giải phương trình đệ quy

- **Lưu ý khi giải một phương trình đệ quy cụ thể**:

(1) Xem phương trình có **dạng đệ quy tổng quát** không ?

(2) Nếu **có**, xem hàm tiến triển d(n) có **dạng hàm nhân** không?

a) Nếu **có**: xác định a, d(b); so sánh a, d(b) để vận dụng một trong ba trường hợp trên.

b) Nếu **không** : tính trực tiếp nghiệm để so sánh.

(3) Suy ra nghiệm của phương trình = Max(NTN,NR)

Ví dụ 1. GPT với T(1) = 1 và

$$1/\ T(n) = 4T\left(\frac{n}{2}\right) + n$$

- Phương trình đã cho có dạng phương trình tổng quát.
- d(n) = n là hàm nhân.
- a = 4 và b = 2.
- d(b) = b = 2 < a =4 (Trường hợp 1)

$\rightarrow$ $T(n) = O(n^{\log_b a}) = O(n^{\log 4}) = \mathbf{O(n^2)}$

Ví dụ 2. GPT với T(1) = 1 và

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

- Phương trình đã cho có dạng phương trình tổng quát.
- $d(n) = n^2$ là hàm nhân.
- $a = 4$ và $b = 2$.
- $d(b) = b^2 = 4 = a$ (Trường hợp 3)

$\rightarrow T(n) = O(n^{\log_b a}\log_b n) = O(n^{\log 4}\log n) = \mathbf{O(n^2\log n)}$

Ví dụ 3. GPT với T(1) = 1 và

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

- Phương trình đã cho có dạng phương trình tổng quát.
- $d(n) = n^3$ là hàm nhân.
- $a = 4$ và $b = 2$.
- $d(b) = b^3 = 8 > a = 4$ (Trường hợp 2)

$\rightarrow T(n) = O(n^{\log_b d(b)}) = O(n^{\log 8}) = \mathbf{O(n^3)}$

Ví dụ 4. GPT với T(1) = 1 và

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

- PT thuộc dạng phương trình tổng quát nhưng **d(n) = nlogn** *không phải là một hàm nhân.*
- a = 2 và b = 2
- NTN = $O(n^{\log_b a}) = O(n^{\log 2}) = $ **O(n)**
- Do d(n) = nlogn không phải là hàm nhân nên ta phải tính nghiệm riêng bằng cách xét trực tiếp

Ví dụ 4. (tt)

$$NR = \sum_{j=0}^{k-1} a^j \boxed{d\left(b^{k-j}\right)} = \sum_{j=0}^{k-1} 2^j \boxed{2^{k-j}\log 2^{k-j}}$$

Do **d(n) = nlogn**

$\rightarrow d(b^{k-j}) = b^{k-j}\log b^{k-j} = 2^{k-j}\log 2^{k-j}$

$$= 2^k \sum_{j=0}^{k-1} (k - j) = 2^k \frac{k(k+1)}{2} = O(2^k k^2)$$

- Theo giả thiết giải phương trình đệ quy tổng quát thì $\mathbf{n = b^k}$ nên $k = \log_b n$, ở đây do b = 2 nên $2^k = n$ và k = logn,
- $NR = O(2^{\log n}\log^2 n) = \mathbf{O(n\log^2 n)} > NTN = O(n)$

$\rightarrow$ $T(n) = \mathbf{O(n\log^2 n)}$

Một số tổng thông dụng

$S = 1 + 2 + 3 + \ldots + n = n(n+1)/2 \approx n^2/2$

$S = 1 + 2^2 + 3^2 + \ldots + n^2 = n(n+1)(2n+1)/6 \approx n^3/3$

$S = 1 + a + a^2 + a^3 + \ldots + a^n = (a^{n+1} - 1)/(a-1)$

Nếu $0 < a < 1$ thì

$S \le 1/(1-a)$

và khi $n \to \infty$ thì

S tiến về $1/(1-a)$

$S = 1 + 1/2 + 1/3 + \ldots + 1/n = \ln(n) + \gamma$

Hằng số Euler $\gamma \approx 0.577215665$

$S = 1 + 1/2 + 1/4 + 1/8 + \ldots + 1/2^n + \ldots \approx 2$

CANTHO UNIVERSITY

Bài tập 4-1. GPT với T(1) = 1 và

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

- Phương trình đã cho có dạng phương trình tổng quát.
- d(n) =1 là hàm nhân.
- a = 1 và b = 2.
- d(b) = 1 = a : Trường hợp 3

→ $T(n) = O(n^{\log_b a}\log_b n) = O(n^{\log 1}\log n) = \mathbf{O(\log n)}$.

Bài tập 4-2. GPT với T(1) = 1 và

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

- Phương trình đã cho có dạng phương trình tổng quát.
- d(n)=logn không phải là hàm nhân.
- a = 2 và b = 2
- NTN = $O(n^{\log_b a}) = O(n^{\log 2}) = O(n)$.
- Tính trực tiếp nghiệm riêng.

Bài tập 4-2. (tt)

$$NR = \sum_{j=0}^{k-1} a^j d(b^{k-j}) = \sum_{j=0}^{k-1} 2^j \log 2^{k-j}$$

$$NR = \sum_{j=0}^{k-1} 2^j (k-j) = \sum_{j=0}^{k-1} k2^j - \sum_{j=0}^{k-1} j2^j$$

$$NR = O(k\sum_{j=0}^{k-1} 2^j) = O(k\frac{2^k - 1}{2-1})$$

$$NR = O(k2^k) = O(n\log n) > n = NTN$$

$$T(n) = O(n\log n)$$

Bài tập 8.

$$C^k_n = \begin{cases} 1 & neu\ k = 0\ hoac\ k = n \\ C^{k-1}_{n-1} + C^k_{n-1} & \end{cases}$$

- Gọi T(n) là thời gian để tính C^k_n
- Thì thời gian để tính C^k_{n-1} là T(n-1)
- Khi n=1 thì k=0 hoặc k=1 => chương trình trả về giá trị 1, tốn O(1) = C_1
- Khi n>1, trong trường hợp xấu nhất, chương trình phải làm các việc:
  - Tính C^k_{n-1} và C^{k-1}_{n-1} : tốn 2T(n-1).
  - Thực hiện phép cộng và trả kết quả: tốn C_2

a) Ta có phương trình: T(1)=C_1 và T(n)=2T(n-1) +C_2

Bài tập 8. (tt)

b) Giải phương trình $T(n) = 2T(n-1) + C_2$

- $T(n) = 2[2T(n-2)+C_2] + C_2$

  $= 4T(n-2) + 3C_2$

  $= 4[2T(n-3) + C_2] + 3C_2$

  $= 8T(n-3) + 7C_2$

  ……

  $= 2^i T(n-i) + (2^i-1) C_2$

- $T(n) = 2^{n-1}C_1 + (2^{n-1}-1) C_2$

  $= (C_1+ C_2)2^{n-1} - C_2 = O(2^n)$