□

Ví dụ 5.69. Trình bày các thuật toán tính a^n với $a \in \mathbb{R}$, $n \in \mathbb{N}$, và đánh giá độ phức tạp tương ứng.

Giải. **Thuật toán 1:**

```
Data: a, n
Result: a^n
1 x = 1                          // x lưu a^1, a^2, ..., a^n
2 for i = 1 to n do
3     x = x * a
```

Vòng lặp **for** ở dòng 2 thực hiện n chu trình, nên thuật toán có độ phức tạp $f(n) = n \in O(n)$, là độ phức tạp tuyến tính.

Thuật toán 2: phương pháp chia đôi để tính lũy thừa

```
1 x = 1
2 i = n
3 while i > 0 do
4     if i lẻ then x = x * a
5     a = a * a
6     i = ⌊i/2⌋
```

Chẳng hạn, tính a^{10}

Bước	x	a	i
Khởi tạo	1	a	10
Dòng 3, chu trình 1	1	a^2	**5**
Dòng 3, chu trình 2	a^2	a^4	2
Dòng 3, chu trình 3	a^2	a^8	**1**
Dòng 3, chu trình 4	a^{10}	a^{16}	$\boxed{0}$

Đặt $f(n)$ là số chu trình của của vòng lặp **while**. Ta chứng minh $f(n) \leq 1+\log_2 n$, $\forall n \in \mathbb{Z}^+$ bằng quy nạp.

1) $f(1) = 1 \leq 1 + \log_2 1$: đúng.

2) Giả sử $f(k) \leq 1 + \log_2 k,\ \forall k = \overline{1, n}$. Ta sẽ chứng minh $f(n+1) \leq 1 + \log_2(n+1)$. Xét thuật toán với đầu vào $n+1$. Sau bước đầu tiên, $i = \lfloor \frac{n+1}{2} \rfloor$, từ đó $f(n+1) = 1 + f\left(\lfloor \frac{n+1}{2} \rfloor\right)$. Mặt khác, $1 \leq \frac{n+1}{2} \leq n$ nên $1 \leq \lfloor \frac{n+1}{2} \rfloor \leq n$, suy ra

$$f\left(\lfloor \frac{n+1}{2} \rfloor\right) \leq 1 + \log_2 \lfloor \frac{n+1}{2} \rfloor \leq 1 + \log_2 \frac{n+1}{2} = \log_2 (n+1).$$

Do đó $f(n+1) \leq 1 + \log_2 (n+1)$.

Vậy $f(n) \leq 1 + \log_2 n \in O(\log_2 n)$, thuật toán có độ phức tạp loga.

□

Cỡ đầu vào n	Độ phức tạp					
	$\log_2 n$	n	$n \log_2 n$	n^2	2^n	$n!$
2	1	2	2	4	4	2
16	4	16	64	256	6.5×10^4	2.1×13
64	6	64	384	4096	1.84×10^{19}	$> 10^{89}$

Cho tập A cỡ n. Xét hai thuật toán:

1) Tìm các tập con cỡ 1 của A. Có n tập con.

2) Tìm tất cả tập con của A. Có 2^n tập con.

Giả sử máy tính xác định mỗi tập con của A mất một nano giây (10^{-9}s). Khi đó nếu $|A| = 64$, thuật toán (1) thực thi xong gần như ngay lập tức, trong 64 nano giây. Tuy nhiên, thuật toán (2) thực hiện trong

$$1.84 \times 10^{19} \text{ nano giây } = 2.14 \times 10^5 \text{ ngày } = 585 \text{ năm}.$$

Bài tập bổ sung

5.38. Ước tính mất bao lâu để phân tích nguyên tố cho một số có 1000 chữ số bằng phép chia thử. Giả sử rằng ta thử tất cả các ước có thể đến căn bậc hai của số đó và có thể thực hiện một triệu tỷ phép chia thử mỗi giây (cỡ siêu máy tính). Chọn một đơn vị thời gian hợp lý cho câu trả lời.

ii) $N(c_1)$ là số nghiệm nguyên không âm của $x_1 + x_2 + x_3 + x_4 = 25$, $x_4 \geq 3$ sao cho $x_1 \geq 4$, bằng $\binom{4+(25-3-4)-1}{25-3-4} = 1330$. Tương tự $N(c_2) = \binom{4+(25-3-6)-1}{25-3-6} = 969$, $N(c_3) = \binom{14}{11} = 364$. Ta có

$$N_1 = N(c_1) + N(c_2) + N(c_3) = 2663.$$

iii) $N_2 = N(c_1c_2) + N(c_1c_3) + N(c_2c_3) = \binom{15}{12} + \binom{10}{7} + \binom{8}{5} = 631.$

iv) $N_3 = N(c_1c_2c_3) = \binom{4}{1} = 4.$

Như vậy, $N(\overline{c_1}\ \overline{c_2}\ \overline{c_3}) = 2300 - 2663 + 631 - 4 = 264.$

□

Định nghĩa 7.1. *Cho số nguyên dương n. Hàm Euler phi, ký hiệu $\Phi(n)$, là số các số nguyên từ 1 tới n và nguyên tố cùng nhau với n.*

Chẳng hạn, $\Phi(2) = 1$, $\Phi(3) = 2$, $\Phi(4) = 2$, $\Phi(5) = 4$, $\Phi(6) = 2$.

```python
from sympy import *
totient(6)  # → 2
```

Nếu p nguyên tố, thì $\Phi(p) = p - 1$. Tổng quát

Cho số nguyên dương $n \geq 2$. Theo định lý cơ bản của số học, n có phân tích $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ trong đó p_i là số nguyên tố, $e_i \in \mathbb{Z}^+$, $1 \leq i \leq k$. Khi đó

$$\Phi(n) = n \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right).$$

Chứng minh. Với phân tích nguyên tố này của n, một số nguyên dương m nguyên tố cùng nhau với n nếu p_i không là ước m, $1 \leq i \leq k$.

Trong các số m từ 1 tới n xét điều kiện

$$c_i: \quad p_i \text{ là ước của } m.$$

và cần tính

$$N(\overline{c_1}\ \overline{c_2} \cdots \overline{c_k}) = N_0 - N_1 + N_2 - N_3 + \cdots + (-1)^k N_k$$

trong

i) $N_0 = n$

ii) $N_1 = \sum_{1\le i\le k} N(c_i) = \sum_{1\le i\le k} \lfloor \frac{n}{p_i} \rfloor = \sum_{1\le i\le k} \frac{n}{p_i}$

iii) $N(c_ic_j)$, $1 \le i < j \le k$, là số các số từ 1 tới n là bội của p_i và p_j, tức là bội của $\mathrm{lcm}(p_i, p_j)$. Mặt khác, p_i, p_j là các số nguyên tố khác nhau, nên $\mathrm{lcm}(p_i, p_j) = p_ip_j$. Suy ra $N(c_ic_j) = \lfloor \frac{n}{p_ip_j} \rfloor = \frac{n}{p_ip_j}$. Ta có

$$N_2 = \sum_{1\le i<j\le k} N(c_ic_j) = \sum_{1\le i<j<k} \frac{n}{p_ip_j}$$

iv) Tương tự

$$\begin{aligned}
N_3 &= \sum_{1\le i<j<l\le k} N(c_ic_jc_l) = \sum_{1\le i<j<l<k} \frac{n}{p_ip_jp_l}, \ldots \\
N_r &= \sum_{1\le i_1<i_2<\cdots<i_r\le k} N(c_{i_1}c_{i_2}\cdots c_{i_r}) = \sum_{1\le i_1<i_2<\cdots<i_r\le k} \frac{n}{p_{i_1}p_{i_2}\cdots p_{i_r}}, \ldots \\
N_k &= N(c_1c_2\cdots c_k) = 1 = \frac{n}{p_1p_2\cdots p_k}
\end{aligned}$$

Các số hạng này có thừa số chung là n, nên

$$\begin{aligned}
N(\overline{c_1}\,\overline{c_2}\cdots\overline{c_k}) &= n\Big(1 - \sum_{1\le i\le k} \frac{1}{p_i} + \sum_{1\le i<j\le k} \frac{1}{p_ip_j} - \sum_{1\le i<j<l\le k} \frac{1}{p_ip_jp_l} + \cdots + \\
&\quad + (-1)^{r-1} \sum_{1\le i_1<i_2<\ldots<i_r\le k} \frac{1}{p_{i_1}p_{i_2}\cdots p_{i_r}} + \cdots + (-1)^k \frac{1}{p_1p_2\cdots p_k}\Big) \\
&= n\sum_{i=1}^{k} \left(1 - \frac{1}{p_i}\right)
\end{aligned}$$

□

Trong Phần 5.4, ta thừa nhận trước công thức đếm số toàn ánh. Bây giờ ta sẽ chứng minh công thức đó.

Số toàn ánh từ tập A cỡ m vào B cỡ n là

$$\begin{aligned}
&\sum_{k=0}^{n} (-1)^k \binom{n}{n-k} (n-k)^m = \sum_{k=0}^{n-1} (-1)^k \binom{n}{n-k} (n-k)^m \\
&= \binom{n}{n} n^m - \binom{n}{n-1} (n-1)^m + \binom{n}{n-2} (n-2)^m - \cdots + (-1)^{n-2} \binom{n}{2} 2^m + \\
&\quad + (-1)^{n-1} \binom{n}{1} 1^m.
\end{aligned}$$

```
( (x**2+x)/(1-x)**3 + x/(1-x)**2 ).simplify()       # hoặc tính trực tiếp:
Sum( (i**2+i)* x**i, (i,0,oo) ).doit().simplify()   #
```

$$\begin{cases} -\dfrac{2x}{(x-1)^3} & \text{for } |x| < 1 \\ \sum_{i=0}^{\infty} ix^i (i+1) & \text{otherwise} \end{cases}$$

Ví dụ 8.16. Với $f(x) = \dfrac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots$ và $g(x) = \dfrac{1}{1+x} = 1 - x + x^2 - x^3 + \cdots$, ta có

$$f(x)g(x) = \frac{1}{1-x}\frac{1}{1+x} = \frac{1}{1-x^2} = 1 + x^2 + x^4 + x^6 + \cdots$$

Do đó, dãy 1, 0, 1, 0, 1, 0, ... là tích chập của hai dãy 1, 1, 1,... và 1, −1, 1, −1, ...

Ví dụ 8.17. Chứng minh $\dbinom{2n}{n} = \sum_{i=0}^{n} \dbinom{n}{i}^2$, $\forall n \in \mathbb{Z}^+$.

Giải. Ta có

$$(1+x)^{2n} = [(1+x)^n]^2 = \left[\sum_{i=0}^{n} \binom{n}{i} x^i\right]^2,$$

nên hệ số của x^n ở hai vế bằng nhau, tức là

$$\binom{2n}{n} = \sum_{0 \le i \le n,\ i+j=n} \binom{n}{i}\binom{n}{j} = \sum_{i=0}^{n} \binom{n}{i}\binom{n}{n-i} = \sum_{i=0}^{n} \binom{n}{i}^2.$$

□

Số Catalan* c_n là số cách tính tích các ma trận $A = A_0A_1A_2 \cdots A_n$.

Ta có các cách đặt dấu ngoặc sau A_k, với $k = \overline{0, n-1}$:

$$A = \underbrace{(A_0A_1A_2 \ldots A_k)}_{c_k \text{ cách}} \underbrace{(A_{k+1}A_{k+2} \ldots A_n)}_{c_{n-k-1} \text{ cách}} \Rightarrow c_n = \sum_{k=0}^{n-1} c_k c_{n-k-1},$$

với $c_1 = 1 = c_0^2 \Rightarrow c_0 = 1$.

Ví dụ 8.18 $(*)$**.** Dùng hàm sinh và phép tính tích chập để tính số Catalan c_n.

Giải. Xét $f(x) = \sum_{n=0}^{\infty} c_n x^n$, ta có $[f(x)]^2 = \sum_{n=0}^{\infty} a_n x^n$ với $a_n = \sum_{k=0}^{n} c_k c_{n-k} = c_{n+1}$. Suy ra

$$[f(x)]^2 = \sum_{n=0}^{\infty} c_{n+1} x^n = \frac{1}{x} \sum_{n=0}^{\infty} c_{n+1} x^{n+1} = \frac{1}{x} \sum_{n=1}^{\infty} c_n x^n = \frac{1}{x}[f(x) - 1]$$

*Engène Charles Catalan, 1814-1894, nhà toán học Bỉ

$$\Rightarrow x[f(x)]^2 - f(x) + 1 = 0 \Rightarrow f(x) = \frac{1 \pm \sqrt{1-4x}}{2x}.$$

Hệ số của x^n trong $f(x)$, tương ứng với mỗi nghiệm, là

$$c_n = \pm\frac{1}{2}\binom{\frac{1}{2}}{n+1}(-4)^{n+1} = \pm\frac{1}{2}\frac{\frac{1}{2}(\frac{1}{2}-1)(\frac{1}{2}-2)\cdots(\frac{1}{2}-n)}{(n+1)!}(-4)^{n+1}$$

Ta chọn nghiệm $f(x)$ ứng với dấu $-$ để hệ số này dương. Cuối cùng,

$$\begin{aligned} c_n &= \frac{1\cdot 3\cdot 5\cdots(2n-1)}{(n+1)!}2^n = \frac{1\cdot 2\cdot 3\cdot 4\cdot 5\cdots(2n-1)(2n)}{2\cdot 4\cdots(2n)}\frac{2^n}{n!(n+1)} \\ &= \frac{1}{n+1}\frac{(2n)!}{n!n!} = \frac{1}{n+1}\binom{2n}{n}. \end{aligned}$$

□

Bài tập 8.2

8.6. Lập và tính hàm sinh cho dãy sau

a) $\binom{8}{0}, \binom{8}{1}, \binom{8}{2}, \ldots, \binom{8}{8}$

b) $\binom{8}{1}, 2\binom{8}{2}, 3\binom{8}{3}, \ldots, 8\binom{8}{8}$

c) $1, -1, 1, -1, 1, -1, \ldots$

d) $0, 0, 0, 6, -6, 6, -6, \ldots$

e) $1, 0, 1, 0, 1, 0, \ldots$

f) $0, 0, 1, a, a^2, a^3, \ldots, a \neq 0$

8.7. Xác định dãy sinh bởi các hàm

a) $f(x) = (2x-3)^3$

b) $f(x) = \dfrac{x^4}{1-x}$

c) $f(x) = \dfrac{x^3}{1-x^2}$

d) $f(x) = \dfrac{1}{1+3x}$

e) $f(x) = \dfrac{1}{3-x}$

f) $f(x) = \dfrac{1}{1-x} + 3x^7 - 11$

8.8. Cho $f(x)$ là hàm sinh dãy $a_0, a_1, a_2, \ldots$, $g(x)$ là hàm sinh dãy $b_0, b_1, b_2, \ldots$ Biểu diễn $g(x)$ theo $f(x)$ nếu

a) $b_3 = 3; b_n = a_n,\ n \in \mathbb{N},\ n \neq 3$

b) $b_3 = 3,\ b_7 = 7; b_n = a_n,\ n \in \mathbb{N},\ n \neq 3, 7$

c) $b_1 = 1,\ b_3 = 3; b_n = 2a_n,\ n \in \mathbb{N},\ n \neq 1, 3$

d) $b_1 = 1,\ b_3 = 3,\ b_7 = 7; b_n = 2a_n + 5,\ n \in \mathbb{N},\ n \neq 1, 3, 7$

8.9. Tìm hệ số tự do, hay số hạng là hằng số, trong $\left(3x^2 - \frac{2}{x}\right)^{15}$.

```
rsolve( -a(n) + (n-1)+a(n-1) , a(n) , {a(1): 0}).simplify
    ()
```

Ta mô tả chi tiết thuật toán với dãy $x = (7, 9, 2, 5, 8)$ bởi hình sau

$i = 0$	x_0	7		7		7		7	$j = 1$	2
	x_1	9		9		9	$j = 2$	2		7
	x_2	2		2	$j = 3$	2		9		9
	x_3	5	$j = 4$	5		5		5		5
	x_4	8		8		8		8		8
	Bốn phép so sánh và hai phép đổi chỗ									
$i = 1$	x_0	2		2		2		2		
	x_1	7		7		7	$j = 2$	5		
	x_2	9		9	$j = 3$	5		7		
	x_3	5	$j = 4$	5		9		9		
	x_4	8		8		8		8		
	Ba phép so sánh và hai phép đổi chỗ									
$i = 2$	x_0	2		2		2				
	x_1	5		5		5				
	x_2	7		7	$j = 3$	7				
	x_3	9	$j = 4$	8		8				
	x_4	8		9		9				
	Hai phép so sánh và một phép đổi chỗ									
$i = 3$	x_0	2								
	x_1	5								
	x_2	7								
	x_3	8	$j = 4$							
	x_4	9								
	Một phép so sánh và không có phép đổi chỗ									

Ví dụ 9.5. Đặt a_n là số hoán vị của n vật, đánh số từ 1 tới n. Lập hệ thức đệ quy và giải a_n.

Giải. Từ mỗi hoán vị của $n - 1$ vật $1, 2, \ldots, n - 1$, ta tạo ra hoán vị của n vật bằng cách xếp vật thứ n vào trước, sau, hoặc chèn vào giữa hoán vị của $n - 1$ vật này. Như vậy, có n vị trí để xếp vật thứ n. Mặt khác, theo định nghĩa, số hoán vị của $n - 1$ vật là a_{n-1}, nên theo quy tắc nhân $a_n = na_{n-1}$, $\forall n \geq 2$. Với $a_1 = 1$, ta tìm được $a_n = n!$. □

Chẳng hạn, cách sinh hoán vị của $\{1, 2\}$ từ $\{1\}$:

	2	1		
		1	2	

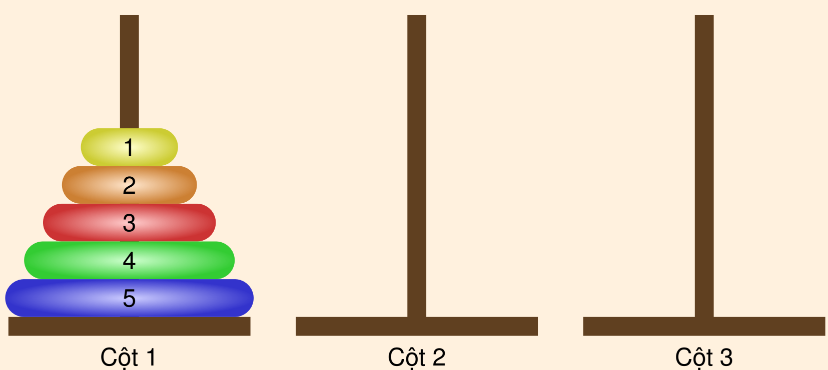
và các hoán vị của $\{1, 2, 3\}$ từ $\{1, 2\}$:

3	1		2	
	1	3	2	
	1		2	3
3	2		1	
	2	3	1	
	2		1	3

```python
def permutations(n):
  if n == 1:
    return [[1]]
  A = []
  for a in permutations(n-1):
    for i in range(n):
        b = a.copy()
        b.insert(i, n)
        A.append(b)
  return A

permutations(3)  # → [[3, 2, 1], [2, 3, 1], [2, 1, 3], [3, 1, 2],
    [1, 3, 2], [1, 2, 3]]
```

Ví dụ 9.6 (Tháp Hà Nội)**.** Xét n đĩa tròn (có đường kính khác nhau) có lỗ ở tâm được xếp chồng lên nhau trên như hình dưới. Trong hình, $n = 5$ và các đĩa xếp ở cột 1 mà đĩa trên nhỏ hơn đĩa dưới. Việc di chuyển các đĩa từ cột này sang cột kia phải thỏa mãn: (1) mỗi lần chỉ chuyển một đĩa, và (2) tại mỗi cột, đĩa trên nhỏ hơn đĩa dưới. Dùng các cột 1, 2, và 3 làm vị trí tạm thời cho các đĩa, ta cần chuyển các hết đĩa sang cột 3.



Đặt a_n là số lần chuyển *ít nhất* để chuyển n đĩa từ cột 1 sang cột 3. Lập hệ thức đệ quy và

Giải. Với $n \geq 3$, xét cột đầu tiên của bàn cờ $2 \times n$. Có thể phủ cột này theo hai cách.

1) Bằng một domino dọc: phần còn lại, là bàn cờ $2 \times (n-1)$, có a_{n-1} cách phủ.

2) Bằng hai domino ngang để phủ cả hai cột đầu bên trái: phần còn lại, là bàn cờ $2 \times (n-2)$, có a_{n-2} cách phủ.

Theo quy tắc cộng, $a_n = a_{n-1} + a_{n-2}, \ \forall n \geq 3$, trong đó $a_1 = 1, a_2 = 2$. Giải hệ thức này, được

$$\begin{aligned}
a_n &= \Big(\frac{1}{2} - \frac{\sqrt{5}}{2}\Big)^n\Big(\frac{1}{2} - \frac{\sqrt{5}}{10}\Big) + \Big(\frac{1}{2} + \frac{\sqrt{5}}{2}\Big)^n\Big(\frac{\sqrt{5}}{10} + \frac{1}{2}\Big)\\
&= \Big(\frac{1+\sqrt{5}}{2}\Big)^n\Big(\frac{1}{2\sqrt{5}} + \frac{1}{2}\Big) - \Big(\frac{1-\sqrt{5}}{2}\Big)^n\Big(\frac{1}{2\sqrt{5}} - \frac{1}{2}\Big)\\
&= \frac{1}{\sqrt{5}}\Big[\Big(\frac{1+\sqrt{5}}{2}\Big)^{n+1} - \Big(\frac{1-\sqrt{5}}{2}\Big)^{n+1}\Big],
\end{aligned}$$

□

Trong ví dụ trên $a_n = F_{n+1}$.

Sử dụng tính chất của số Fibonacci [có thể chứng minh bằng nguyên lý quy nạp],

$$F_n > \alpha^{n-2}, \ \forall n \geq 3, \ \text{với } \alpha = \frac{1+\sqrt{5}}{2},$$

Gabriel Lamé* đã chứng minh

Ví dụ 9.15 (Định lý Lamé)**.** Cho $a, b \in \mathbb{Z}^+$, $a, b \geq 2$. Số phép chia dùng trong thuật toán Euclid để tìm ước chung lớn nhất của a và b không quá 5 lần số chữ số của b.

Giải. Đặt $r_0 = a$ và $r_1 = b$, ta có

$$\begin{aligned}
r_0 &= r_1 q_1 + r_2, && 0 < r_2 < r_1\\
r_1 &= r_2 q_2 + r_3, && 0 < r_3 < r_2\\
r_2 &= r_3 q_3 + r_4, && 0 < r_4 < r_3\\
&\dots\dots\dots\dots\dots\dots\\
r_{n-2} &= r_{n-1} q_{n-1} + r_n, && 0 < r_n < r_{n-1}\\
r_{n-1} &= r_n q_n.
\end{aligned}$$

Khi đó, $\gcd(a, b) = r_n$, là phần dư khác không cuối cùng, và thuật toán thực hiện n phép chia.

*Gabriel Lamé, 1795–1870, nhà toán học Pháp

Ta thấy, $q_i \geq 1,\ \forall i = \overline{1, n}$. Riêng $q_n \geq 2$, vì $r_{n-1} = r_n q_n$ mà $0 < r_n < r_{n-1}$. Như vậy

$$\begin{aligned}
r_n > 0 \Rightarrow r_n &\geq 1 = F_2 \\
r_{n-1} = r_n q_n &\geq 1 \cdot 2 = 2 = F_3 \\
r_{n-2} = r_{n-1} q_{n-1} + r_n &\geq F_3 \cdot 1 + F_2 = F_4 \\
r_{n-3} = r_{n-2} q_{n-2} + r_{n-1} &\geq F_4 \cdot 1 + F_3 = F_5 \\
&\dots\dots\dots\dots \\
r_2 = r_3 q_3 + r_4 &\geq F_{n-1} \cdot 1 + F_{n-2} = F_n \\
b = r_1 = r_2 q_2 + r_3 &\geq F_n \cdot 1 + F_{n-1} = F_{n+1}
\end{aligned}$$

Dẫn đến

$$\begin{aligned}
b &\geq F_{n+1} > \alpha^{(n+1)-2} = \alpha^{n-1} \\
\Rightarrow n - 1 &< \log_\alpha b = \log_\alpha 10 \cdot \log_{10} b = 4.784971 \log_{10} b < 5 \log_{10} b.
\end{aligned}$$

Nếu b có k chữ số, thì $10^{k-1} \leq b < 10^k$, nên $\log_{10} b < k$. Do đó $n - 1 < 5k$, hay $n \leq 5k$, tức là số phép chia trong thuật toán Euclid không quá 5 lần số chữ số của b. □

Ví dụ 9.16. Tìm hệ thức đệ quy của a_n, là số xâu nhị phân độ dài n không có các số 0 liên tiếp.

Giải. **Cách 1:** Với mỗi xâu đếm bởi a_n, có hai khả năng:

1) Số đầu là 1, thì phần còn lại là xâu độ dài $n - 1$ không có các số 0 liên tiếp. Số xâu như vậy là a_{n-1}.
2) Số đầu là 0, thì số thứ hai phải là 1, và phần còn lại là xâu độ dài $n - 2$ không có các số 0 liên tiếp. Số các xâu như vậy là a_{n-2}.

Theo quy tắc cộng, $a_n = a_{n-1} + a_{n-2},\ \forall n \geq 3$. Ta xác định thêm $a_1 = 2$, $a_2 = 3$.

Cách 2: Cách này sử dụng các biến phụ. Trong các xâu đếm bởi a_n, đặt $a_n^{(0)}$ là số xâu số đầu là 0, và $a_n^{(1)}$ là số xâu có số đầu là 1. Khi đó $a_n = a_n^{(0)} + a_n^{(1)}$. Vì mỗi xâu dạng $1s$ đếm bởi $a_n^{(1)}$ khi và chỉ khi xâu s đếm bởi a_{n-1}, nên $a_n^{(1)} = a_{n-1}$.

Với mỗi xâu đếm bởi a_n, có hai khả năng:

1) Số thứ hai là 0, thì số đầu chỉ có thể là 1. Số xâu như vậy là $a_{n-1}^{(0)}$.
2) Số thứ hai là 1, thì số đầu có hai lựa chọn, là 0 hoặc 1. Số xâu như vậy là $2a_{n-1}^{(1)}$.

Xét $n = 2^k \Leftrightarrow k = \log_2 n$, và $a_k = f(n) = f(2^k)$, thì

$$a_k = 2 + f(2^{k-1}) = 2 + a_{k-1}$$

trong đó $a_0 = f(1) = 2$ (dòng 2, 3). Suy ra $a_k = 2 + 2k$, tức là $f(n) = 2 + 2\log_2 n \in O(\log_2 n)$. Thuật toán có độ phức tạp loga. □

Ví dụ 9.32 (Sắp xếp trộn)**.** Sắp xếp dãy $a_1, a_2, \ldots, a_n$ thành dãy tăng dần: trình bày thuật toán chia để trị và đánh giá độ phức tạp của thuật toán.

Giải. Các bước của thuật toán:

1) Nếu dãy chỉ có 1 số, ta không cần làm gì cả. Ngược lại, chuyển xuống thực hiện bước (2).

2) Chia đôi dãy được hai dãy con $a_1, a_2, \ldots, a_k$ và $a_{k+1}, a_{k+2}, \ldots, a_n$ cỡ $\frac{n}{2}$.

   Sắp xếp hai dãy này, được hai dãy tăng.

3) Trộn hai dãy tăng này cho thành một dãy tăng.

Để thực hiện bước (3), ta dùng thuật toán dưới đây. Trong thuật toán này, để trộn hai dãy tăng cỡ m và n, số chu trình tối giản là $m + n$.

```
def merge(a, b):
  n, m = len(a), len(b)
  c = [0] * (n + m)
  i, j, k = 0, 0, 0
  while i < n and j < m:
    if a[i] < b[j]:
      c[k] = a[i]
      i += 1
    else:
      c[k] = b[j]
      j += 1
    k += 1

  while i < n:
    c[k] = a[i]
    i += 1
    k += 1
  while j < m:
    c[k] = b[j]
    j += 1
```

```
    k += 1
  return c

merge([2, 3], [1, 4, 5])  # → 1, 2, 3, 4, 5
```

Chương trình chính như sau:

```
1 def merge_sort(a):
2   n = len(a)
3   if n == 1:
4     return a
5   k = n // 2
6   L = merge_sort(a[:k])       # sắp xếp dãy cỡ n/2
7   R = merge_sort(a[k:])       # như dòng 6
8   return merge(L, R)          # n/2 + n/2 = n chu trình tối giản
9 merge_sort([1, 5, 3, 4, 2])  # → 1, 2, 3, 4, 5
```

Gọi $f(n)$ là số chu trình tối giản của thuật toán với dãy cỡ n. Ta có

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

Xét $n = 2^k \Leftrightarrow k = \log_2 n$, và $a_k = f(n) = f(2^k)$, thì

$$a_k = 2f(2^{k-1}) + 2^k = 2a_{k-1} + 2^k,$$

trong đó $a_0 = f(1) = 0$.

Giải hệ thức đệ quy này, được $a_k = 2^k k$, tức là $f(n) = n \log_2 n$. Thuật toán có độ phức tạp $n \log_2 n$.

□

Dưới đây là một số kết luận, đánh giá về hệ thức chia để trị.

Định lý 9.1. *Cho $a, b, c \in \mathbb{Z}^+$, $b \geq 2$, và $f : \mathbb{Z}^+ \to \mathbb{R}$. Nếu*

$$\begin{aligned} &f(1) = c, \quad \text{và} \\ &f(n) = af\left(\frac{n}{b}\right) + c, \quad \text{với } n = b^k, k \geq 1, \end{aligned}$$

thì với mọi $n = 1, b, b^2, \ldots$

$$f(n) = \begin{cases} c(\log_b n + 1) & \text{nếu } a = 1 \\ \dfrac{c(an^{\log_b a} - 1)}{a - 1} & \text{nếu } a \geq 2. \end{cases}$$