Mục lục

Chương 9

Hệ thức đệ quy

Hệ thức đệ quy xuất hiện khi ta đếm số cách (hoặc, bước) giải (hoặc, lập trình) một bài toán theo phương pháp đệ quy. Nội dung chương tập trung vào việc xây dựng hệ thức đệ quy cho bài toán, còn việc giải hệ thức đệ quy được thực hiện chủ yếu bằng Python. Công thức nghiệm hay phương pháp giải các loại hệ thức đệ quy chỉ được trình bày tổng quát, không có ví dụ minh họa.

9.1 Định nghĩa

Cho dãy $(a_n)_{n\geq 0}$. *Hệ thức đệ quy*[*] của dãy là một công thức biểu diễn sự phụ thuộc của a_n vào các phần tử trước nó $a_{n-1}, a_{n-2}, \dots, a_1, a_0$, và có thể cả n

$$a_n = F\left(a_{n-1}, a_{n-2}, \dots, a_1, a_0, n\right). \tag{9.1}$$

Cho $k \in \mathbb{Z}^+$, hệ thức đệ quy *cấp k* biểu diễn a_n phụ thuộc vào k phần tử trước nó

$$a_n = F\left(a_{n-1}, a_{n-2}, \dots, a_{n-k}, n\right), \quad n = k, k+1, \dots, \tag{9.2}$$

trong đó dãy hoàn toàn xác định nếu biết k giá trị ban đầu $a_0, a_1, \dots, a_{k-1}$. Hệ thức này cũng có thể viết dưới dạng[†]

$$\begin{aligned} a_{n+1} &= F\left(a_n, a_{n-1}, \dots, a_{n-k+1}, n\right), \quad n = k-1, k, \dots, \quad \text{hoặc} \\ a_{n+k} &= F\left(a_{n+k-1}, a_{n+k-2}, \dots, a_n, n\right), \quad n = 0, 1, \dots. \end{aligned}$$

Hệ thức đệ quy gọi là *tuyến tính* nếu có dạng

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f_n, \quad n \geq k, \tag{9.3}$$

trong đó f_n và các *hệ số* $c_1, c_2, \dots, c_k$ là hàm phụ thuộc n, đặc biệt ta hay gặp các hệ số là hằng số. Khi $f_n \equiv 0$, hệ thức gọi là *thuần nhất*, và ngược lại gọi là *không thuần nhất*.

Ví dụ 9.1. Dãy Fibonacci có

a) hệ thức đệ quy $F_n = F_{n-1} + F_{n-2}, \ \ n \geq 2$, là hệ thức đệ quy tuyến tính thuần nhất cấp hai hệ số hằng. Hệ thức này cũng có thể viết dưới dạng

$$\begin{aligned} &F_{n+1} = F_n + F_{n-1}, \ \ n \geq 1, \quad \text{hoặc} \\ &F_{n+2} - F_{n+1} - F_n = 0, \ \ n \geq 0. \end{aligned}$$

b) hai điều kiện ban đầu $F_0 = 0$, và $F_1 = 1$.

Trong một số trường hợp, hệ thức đệ quy cấp k giải được *nghiệm tổng quát*

$$a_n = \varphi\left(n, C_1, C, \dots, C_k\right), \tag{9.4}$$

[*] còn gọi là *phương trình sai phân*

[†] Cấp của hệ thức đệ quy nên được xác định bằng cách lấy hiệu chỉ số của phần tử được biểu diễn với chỉ số nhỏ nhất của phần tử ở vế phải.

trong đó $C_1, C_2, \dots, C_k$ gọi là các *hệ số bất định*. Các hệ số này hoàn toán xác định khi biết k phần tử của dãy, thường là *k điều kiện ban đầu* $a_0, a_1, \dots, a_{k-1}$. Khi đó ta có *nghiệm tường minh*.

Một dãy $\left(a_n^*\right)$ thỏa mãn hệ thức đệ quy gọi là một *nghiệm riêng* của hệ thức đệ quy đó.

9.2 Python

Python có thể giải được hệ thức đệ quy tuyến tính hệ số hằng (9.3) trong một số trường hợp đặc biệt của f_n.

Bước 1: Khai báo thư viện SymPy, biến n, và dãy số a_n như một hàm

```
1 from sympy import *

2 n = symbols('n')
3 a = symbols('a', cls=Function)
```

Bước 2: Biến đổi hệ thức đệ quy để đưa về dạng có một vế bằng 0. Đơn giản nhất là chuyển vế, chẳng hạn

$$(9.3) \Leftrightarrow -a_n + c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f_n = 0, \quad n \geq k. \tag{9.5}$$

Ký hiệu vế trái của hệ thức trên là VT. Trong Python, a_n trong VT được viết là a(n).

Bước 3: Giải hệ thức đệ quy

1. Chưa có điều kiện ban đầu, ta được nghiệm tổng quát

```
4 rsolve(
5     VT,
6     a(n)
7 )
```

2. Với điều kiện ban đầu $a_{i_1} = v_1, a_{i_2} = v_2, \dots, a_{i_k} = v_k$, ta nhập thêm vào sau dòng 6, được nghiệm tường minh

```
4 rsolve(
5     VT,
```

```
6         a(n),
7         {a(i_1): v_1, a(i_2): v_2, ..., a(i_k): v_k}
8   )
```

Lưu ý điều kiện ban đầu ở dòng 7 có thể viết gọn thành

```
7         {i_1: v_1, i_2: v_2, ..., i_k: v_k}
```

Trong trình bày sau này, để ngắn gọn, các lệnh khai báo như dòng 1–3 ở Bước 1 sẽ được lược bớt, hoặc điều chỉnh. Vì vậy, muốn mã đầy đủ và chạy được, ta cần bổ sung lại, hoặc thay thế các khai báo này.

Ví dụ 9.2. Tìm công thức tường minh của dãy Fibonacci F_n.

Kết quả. $F_n = \dfrac{1}{\sqrt{5}}\left[\left(\dfrac{1+\sqrt{5}}{2}\right)^n - \left(\dfrac{1-\sqrt{5}}{2}\right)^n\right]$. □

```python
from sympy import *

n = symbols('n')
F = symbols('F', cls=Function)
rsolve(
    -F(n) + F(n-1) + F(n-2),
    F(n),
    {F(0): 0, F(1): 1}
)  # → -√5(1/2 - √5/2)^n/5 + √5(1/2 + √5/2)^n/5, hình thức hơi khác so với kết quả ghi ở trên
```

Đặt $\alpha = \dfrac{1+\sqrt{5}}{2}$, $\beta = \dfrac{1-\sqrt{5}}{2}$, α gọi là *tỷ lệ vàng*, ta được *dạng Binet*[*] của F_n

$$F_n = \frac{1}{\sqrt{5}}(\alpha^n - \beta^n) = \frac{\alpha^n - \beta^n}{\alpha - \beta}.$$

Trong các phần tiếp theo, ta liệt kê một số dạng hệ thức đệ quy có thể giải được và ứng dụng của chúng.

[*] do Jacques Philippe Marie Binet (1786 – 1856) phát biểu năm 1843

9.3 Hệ thức đệ quy tuyến tính cấp một

Hệ thức đệ quy tuyến tính cấp một

$$a_n = c_n a_{n-1} + f_n \tag{9.6}$$

có nghiệm

$$a_n = c_1 c_2 \cdots c_n \left(a_0 + \frac{f_1}{c_1} + \frac{f_2}{c_1 c_2} + \frac{f_3}{c_1 c_2 c_3} + \cdots + \frac{f_n}{c_1 c_2 \cdots c_n}\right). \tag{9.7}$$

Trường hợp đặc biệt

1. Với $c_n = 1$, $f_n = d$, *cấp số cộng*

$$a_n = a_{n-1} + d \tag{9.8}$$

   có nghiệm

$$a_n = a_0 + nd. \tag{9.9}$$

2. Với $c_n = q$, $f_n = 0$, *cấp số nhân*

$$a_n = qa_{n-1} \tag{9.10}$$

   có nghiệm

$$a_n = a_0 q^n. \tag{9.11}$$

Hầu hết hệ thức (9.6) với c_n không phải hằng số và $f_n \neq 0$ không giải được bằng Python, trừ phi nó có thể tính được tổng trong dấu ngoặc của (9.7).

Hai ví dụ 9.3, 9.4 về toán tài chính.

Ví dụ 9.3 (Bài toán lãi suất)**.** Một ngân hàng trả lãi kép 5% hàng năm cho tài khoản tiết kiệm. Ban đầu, tài khoản có 100 triệu. Đặt a_n (triệu) là số tiền trong tài khoản sau n năm.

a) Xác định a_0, a_1, a_2, a_3.

b) Tìm hệ thức đệ quy và giải a_n.

c) Sau 10 năm, tài khoản có bao nhiêu tiền? Sau bao nhiêu năm, tài khoản vượt gấp đôi.

Giải. a)

$$\begin{aligned} a_0 & & &= 100 \\ a_1 &= 100 + 100 \cdot 5\% & &= 105 \\ a_2 &= 105 + 105 \cdot 5\% & &= 110.25 \\ a_3 &= 110.25 + 110.25 \cdot 5\% & &= 115.7625 \end{aligned}$$

```python
a = 100
for _ in range(3):
    a = a + a * 5 / 100
    print(a)
```

b) Số tiền sau n năm = số tiền năm trước + lãi suất, tức là

$$a_n = a_{n-1} + a_{n-1} \cdot 5\% = 1.05 a_{n-1},$$

trong đó $a_0 = 100$.

Vì (a_n) là cấp số nhân, nên $a_n = 100 \cdot 1.05^n$.

```python
sol = rsolve(
    a(n) - 1.05*a(n-1),
    a(n),
    {a(0): 100}
)
sol
```

c) i) $a_{10} = 100 \cdot 1.05^{10} = 162.889$ (triệu)

```python
sol.subs(n, 10)   # sympy.subs
```

ii) Cần tìm $n \in \mathbb{Z}^+$ nhỏ nhất sao cho

$$a_n \geq 2a_0 \Leftrightarrow 100{\cdot}1.05^n \geq 2{\cdot}100 \Leftrightarrow 1.05^n \geq 2 \Leftrightarrow n \geq \log_{1.05} 2 = 14.20$$

được $n = 15$.

```python
log(2., 1.05)   # sympy.log
```

□

Ví dụ 9.4 (Bài toán trả góp)**.** An vay một khoản tiền S với lãi suất tháng r, và phải trả trong T tháng – không tính tháng vay.

a) Tìm số tiền cố định P mà An phải trả hàng tháng.

b) Áp dụng cho trường hợp cụ thể: $S = 100$ triệu, $r = 1\%/$tháng, $T = 12$ hoặc 24 tháng.

Giải. a) Đặt a_n là số tiền còn nợ vào cuối tháng của lần trả thứ n, bao gồm

i) số tiền còn nợ của tháng trước = a_{n-1}, $n \geq 1$;

ii) lãi = $r \times a_{n-1}$; và

iii) số tiền trả vào cuối tháng = $-P$

tức là

$$a_n = a_{n-1} + ra_{n-1} - P = (1 + r)\, a_{n-1} - P$$

trong đó $a_0 = S$. Giải hệ thức, được

$$a_n = \frac{P}{r} + \frac{(-P + Sr)\,(r + 1)^n}{r} = \frac{P + (-P + Sr)\,(r + 1)^n}{r}$$

```python
from sympy import *

n, P, r, S = symbols('n P r S')
a = symbols('a', cls=Function)
sol = rsolve(
    -a(n) + (1+r) * a(n-1) - P,
    a(n),
    {a(0): S}
)
sol

sol.simplify()
```

An muốn $a_T = 0$, nên

$$P = \frac{Sr(r + 1)^T}{(r + 1)^T - 1} = \frac{Sr}{1 - (1 + r)^{-T}}.$$

```python
sol_P = solve(sol, P)
sol_P

sol_P[0]
```

b) Vay 100 triệu với lãi suất 1%/tháng, muốn trả trong 12 tháng thì mỗi tháng cần trả

$$P = \frac{100 \cdot 0.01}{1 - (1 + 0.01)^{-12}} = 8.884 \text{ (triệu)}$$

```
sol_P[0].subs({S: 100, r: 0.01, n: 12})
```

Tương tự, muốn trả trong 24 tháng, mỗi tháng cần trả 4.707 triệu.

□

Các ví dụ 9.5 đến 9.7 về đánh giá độ phức tạp của thuật toán trong khoa học máy tính.

Ví dụ 9.5. a) Nêu một quy tắc xây dựng phân hạng của n từ phân hạng của $n-1$, với $n = 3, 4$.

b) Đặt a_n là số các phân hạng của $n \in \mathbb{Z}^+$. Lập hệ thức đệ quy và giải a_n.

c) Từ ý tưởng ở ý (a) và (b), viết chương trình liệt kê các phân hạng của $n \in \mathbb{Z}^+$.

d) Đặt a_n là số phép so sánh trong chương trình trên ứng với đầu vào $n \in \mathbb{Z}^+$. Lập hệ thức đệ quy cho (a_n) và tìm công thức tường minh của a_n.

Giải. a) Từ phân hạng của $n = 3$, ta lập phân hạng của $n = 4$ theo hai cách

Cách 1: Cộng 1 vào số hạng đầu của phân hạng, chẳng hạn $2 + 1 \to 3 + 1$.

Cách 2: Chèn thêm số hạng 1 trở thành số hạng đầu tiên của phân hạng, chẳng hạn $3 \to 1 + 3$.

Bảng xây dựng chi tiết

$n = 2$		$n = 3$		$n = 4$	
				(1)	4
				(2)	$3 + 1$
		(1)	3	(3)	$2 + 2$
(1)	2	(2)	$2 + 1$	(4)	$2 + 1 + 1$
(2)	$1 + 1$	(3)	$1 + 2$	(5)	$1 + 3$
		(4)	$1 + 1 + 1$	(6)	$1 + 2 + 1$
				(7)	$1 + 1 + 2$
				(8)	$1 + 1 + 1 + 1$

b) Với $n \geq 2$, phân hạng của n có dạng

$$x_1 + x_2 + \cdots + x_k = n \tag{$*$}$$

trong đó $k, x_1, x_2, \dots, x_k \in \mathbb{Z}^+$. Xét hai trường hợp

1) $x_1 \geq 2$. Khi đó

$$(*) \Leftrightarrow (x_1 - 1) + x_2 + \cdots + x_k = n - 1$$

là phân hạng của $n - 1$, tức là mỗi phân hạng dạng này tương ứng $1 - 1$ với một phân hạng của $n - 1$. Theo định nghĩa, số phân hạng này là a_{n-1}.

2) $x_1 = 1$. Khi đó

$$(*) \Leftrightarrow x_2 + \cdots + x_k = n - 1$$

cũng là phân hạng của $n - 1$, và ta thu được a_{n-1} phân hạng loại này.

Theo quy tắc cộng

$$a_n = a_{n-1} + a_{n-1} = 2a_{n-1}, \quad n \geq 2$$

trong đó $a_1 = 1$. Đây là cấp số nhân nên $a_n = a_1 \cdot 2^{n-1} = 2^{n-1}$, $\forall n \geq 1$.

c) Đặt một số đầu vào đầu ra cho bài toán

Input : $n = 3$
Output : `[[3], [2, 1], [1, 2], [1, 1, 1]]`
Input : $n = 1$
Output : `[[1]]`

```python
def summands(n):
    if n == 1:
        return [[1]]
    S = []
    for s in summands(n-1):
        s[0] += 1
        S.append(s)
    for s in summands(n-1):
        s = [1] + s
        S.append(s)
    return S

summands(3)
```

d)
- Nếu $n = 1$, chương trình chỉ có 1 phép so sánh, và tại dòng 2. Như vậy

$$a_1 = 1. \tag{1}$$

- Nếu $n \geq 2$
  * Dòng 2 có 1 phép so sánh
  * Dòng 5 (và do đó dòng 5 − 7) có a_{n-1} phép so sánh
  * Dòng 8 − 10 có a_{n-1} phép so sánh

  Theo quy tắc cộng

$$a_n = 1 + a_{n-1} + a_{n-1} = 2a_{n-1} + 1. \tag{2}$$

Giải (1) và (2) được

$$a_n = 2^n - 1.$$

□

Ví dụ 9.6. Thuật toán *sắp xếp nổi bọt*

```
1 def BubleSort(x):
2     n = len(x)
3     if n == 1:
4         return x
5     for i in range(n-1, 0, -1):
6         if x[i] < x[i-1]:
7             x[i], x[i-1] = x[i-1], x[i]
8     return [x[0]] + BubleSort(x[1:])

9 BubleSort([7, 9, 2, 5, 8])  # [2, 5, 7, 8, 9]
```

Đặt a_n là số phép so sánh cần dùng để sắp xếp dãy cỡ n. Lập hệ thức đệ quy và giải a_n.

Giải.
- Với $n = 1$, chương trình có đúng 1 phép so sánh, và ở dòng 3:

$$a_1 = 1. \tag{1}$$

- Với $n \geq 2$.
  - Dòng 3 có 1 phép so sánh

– Dòng 6 có 1 phép so sánh ⇒ dòng 5–7 với $i = n-1, n-2, \ldots, 1$ sẽ có $n-1$ phép so sánh

– Xét dòng 8: trong dòng 2, cỡ của `x` là n ⇒ `x[1:]` gồm $x_1, x_2, \ldots, x_{n-1}$ gồm $n-1$ phần tử. Theo định nghĩa, dòng này có a_{n-1} phép so sánh.

Theo quy tắc cộng

$$a_n = 1 + (n-1) + a_{n-1} = a_{n-1} + n \tag{2}$$

Giải (1) và (2) được

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

□

Ví dụ 9.7. Xét chương trình liệt kê hoán vị trong Ví dụ 1.14

```
1  def permutations(a):
2      n = len(a)
3      if n == 1:
4          return [a]
5      P = []
6      for i in range(n):
7          b = a.copy()
8          x = b.pop(i)
9          for p in permutations(b):
10             p = [x] + p
11             P.append(p)
12     return P
```

Lập hệ thức đệ quy và giải a_n khi

a) a_n là số hoán vị của n vật tạo ra bởi chương trình.

b) a_n là số phép so sánh trong chương trình sinh hoán vị của n vật. Từ đó đánh giá độ phức tạp của thuật toán.

Giải. a) – Khi $n = 1$, dòng 3–4 cho ta 1 hoán vị:

$$a_1 = 1. \tag{1}$$

– Khi $n \geq 2$, số hoán vị a_n là cỡ của `P` tại dòng 12

  * Dòng 5: P chưa có gì
  * Dòng 11 thêm một hoán vị mới vào P. Vì dòng 7–8 cho cỡ của b là $n-1$ nên số hoán vị thêm vào P ở dòng 9–11 là a_{n-1}. Theo quy tắc nhân, số hoán vị thêm vào P ở dòng 6–11 là $n \times a_{n-1}$.

  Vậy

$$a_n = na_{n-1}. \tag{2}$$

Giải (1) và (2) được

$$a_n = n!.$$

b)
- Khi $n = 1$, chương trình chỉ có 1 phép so sánh, và ở dòng 3:

$$a_1 = 1. \tag{3}$$

- Khi $n \geq 2$.
  * Dòng 3 có 1 phép so sánh.
  * Dòng 9 có phép so sánh nằm trong hàm `permutations(b)`. Vì b có cỡ $n-1$ nên dòng này (và do đó dòng 7–11) có a_{n-1} phép so sánh. Theo quy tắc nhân, dòng 6–11 có $n \times a_{n-1}$ phép so sánh.

  Theo quy tắc cộng

$$a_n = 1 + na_{n-1} = na_{n-1} + 1. \tag{4}$$

Đây là hệ thức đệ quy tuyến tính cấp 1, áp dụng (9.7) với $c_n = n$, $f_n = 1$. Ta cũng phải tìm a_0 từ (4) bằng cách thay $n = 1$

$$a_1 = 1 \cdot a_0 + 1 \Rightarrow a_0 = 0.$$

Ta có

$$\begin{aligned}
a_n &= c_1 c_2 \cdots c_n \left(a_0 + \frac{f_1}{c_1} + \frac{f_2}{c_1 c_2} + \cdots + \frac{f_n}{c_1 c_2 \cdots c_n} \right) \\
&= 1 \cdot 2 \cdots n \left(0 + \frac{1}{1} + \frac{1}{1 \cdot 2} + \cdots + \frac{1}{1 \cdot 2 \cdots n} \right) \\
&= n! \left(\frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!} \right)
\end{aligned}$$

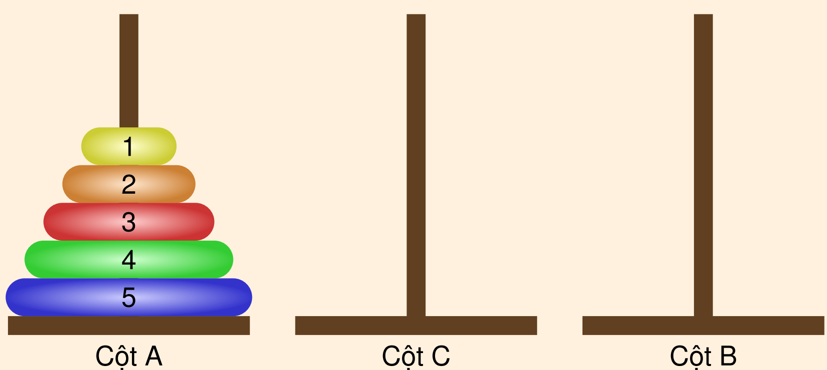
Liên hệ biểu thức trong ngoặc với khai triển Maclaurin

$$\begin{aligned}
e^x &= 1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots \\
\Rightarrow e &= 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!} + \cdots, \quad (\text{cho } x = 1) \\
\Rightarrow & \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!} < e - 1 \\
\Rightarrow & a_n < (e-1)\, n! \\
\Rightarrow & a_n \in O(n!)
\end{aligned}$$

Vậy thuật toán có độ phức tạp giai thừa.

□

Ví dụ 9.8 (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 A, B, và C làm vị trí tạm thời cho các đĩa, ta cần chuyển các hết đĩa sang cột B.



a) Đặt a_n là số lần chuyển *ít nhất* để chuyển n đĩa từ cột 1 sang cột 2. Lập hệ thức đệ quy và giải a_n.

b) Mô tả chi tiết các lần chuyển với $n = 2, 3, 4$ đĩa.

c) Viết chương trình liệt kê các bước chuyển n đĩa.

d) Giả sử một máy tính mô phỏng các lần chuyển, được một tỷ lần chuyển mỗi giây. Ước lượng thời gian máy tính cần chuyển xong hệ 64 đĩa.

Giải. a) Với n đĩa, $n \geq 2$, ta có thể làm như sau:

i) Chuyển $n-1$ đĩa trên cùng từ cột 1 sang cột 3. Việc này cần ít nhất a_{n-1} lần chuyển.

ii) Chuyển đĩa to nhất từ cột 1 sang cột 2. Có một lần chuyển.

iii) Cuối cùng, chuyển $n-1$ đĩa từ cột 3 lên trên đĩa to nhất ở cột 2. Việc này cần ít nhất a_{n-1} lần chuyển.

Do đó

$$a_n \leq a_{n-1} + 1 + a_{n-1} = 2a_{n-1} + 1. \tag{1}$$

Mặt khác, với n đĩa, $n \geq 2$, tại thời điểm nào đó, đĩa to nhất (đĩa dưới cùng tại cột 1) phải được chuyển sang cột 2. Lúc này, cột 2 không có đĩa nào, và $n-1$ đĩa nhỏ hơn đã chuyển sang cột 3. Để chuyển $n-1$ đĩa này, cần ít nhất a_{n-1} lần chuyển. Đĩa to nhất cần di chuyển ít nhất một lần để sang được cột 2. Sau đó, để đặt $n-1$ đĩa nhỏ lên trên đĩa to nhất (đều trên cột 2), cần ít nhất a_n bước nữa. Vậy

$$a_n \geq a_{n-1} + 1 + a_{n-1} = 2a_{n-1} + 1. \tag{2}$$

Từ (1) và (2) suy ra

$$a_n = 2a_{n-1} + 1 \tag{3}$$

trong đó

$$a_1 = 1. \tag{4}$$

Giải hệ thức (3) với điều kiện (4) được

$$a_n = 2^n - 1. \tag{5}$$

b) Khi một đĩa d, $1 \leq d \leq n$, được chuyển từ cột A sang cột B, ta lập bộ $(d, \mathrm{A}, \mathrm{B})$.

$n = 2$**:** (1, A, C)
(2, A, B)
(1, C, B)

$n = 3$**:** (1, A, B) (2, A, C) (1, B, C)
(3, A, B)
(1, C, A) (2, C, B) (1, A, B)

$n = 4$:

(1, A, C) (2, A, B) (1, C, B) (3, A, C) (1, B, A) (2, B, C) (1, A, C) (4, A, B)

(1, C, B) (2, C, A) (1, B, A) (3, C, B) (1, A, C) (2, A, B) (1, C, B)

c) Input : $n = 2$

Output : `[[1, 'A', 'C'], [2, 'A', 'B'], [1, 'C', 'B']]`

Input : $n = 1$

Output : `[[1, 'A', 'B']]`

```
# Chuyển n đĩa từ cột A sang cột B, dùng cột trung gian C
def hanoi_tower(n, A, B, C):
    if n == 1:
        return [[1, A, B]]
    return hanoi_tower(n-1, A, C, B) + [[n, A, B
  ]] + hanoi_tower(n-1, C, B, A)

hanoi_tower(3, 'A', 'B', 'C')
```

d) Theo (5), thời gian máy tính cần để chuyển hệ 64 đĩa

$$\frac{2^{64} - 1}{10^9} \text{ giây } = 18\,446\,744\,074 \text{ giây } = \frac{18\,446\,744\,074}{3600 \times 24 \times 365} \approx 585 \text{ năm!}$$

□

Ví dụ 9.9. Trong Ví dụ 8.24, ta đã đếm số xâu tứ phân độ dài n có một số chẵn các số 1, ký hiệu a_n. Trong phần này ta tính a_n bằng phương pháp đệ quy.

a) Lập hệ thức đệ quy và giải a_n.

b) Áp dụng ý tưởng ở ý (a), viết chương trình liệt kê các xâu như vậy.

Giải. a) Trong các xâu đếm bởi a_n, xét hai trường hợp:

1) Ký tự đầu là 1, thì xâu gồm $n - 1$ ký tự còn lại phải có một số lẻ các số 1. Số xâu như vậy bằng số xâu độ dài $n - 1$, trừ đi số xâu, vẫn độ dài $n - 1$, có một số chẵn các số 1, tức là $4^{n-1} - a_{n-1}$.

2) Ký tự đầu là 0, 2, hoặc 3. Khi đó xâu $n - 1$ ký tự còn lại có một số chẵn các số 1, gồm a_{n-1} xâu. Theo quy tắc nhân, trường hợp này có $3a_{n-1}$ xâu.

Theo quy tắc cộng

$$a_n = (4^{n-1} - a_{n-1}) + 3a_{n-1} = 2a_{n-1} + 4^{n-1}, \ n \geq 2, \quad (1)$$

trong đó các xâu như vậy độ dài 1 là 0, 2, 3:

$$a_1 = 3. \quad (2)$$

Giải (1) và (2), được

$$a_n = \frac{2^n}{2} + \frac{4^n}{2} = 2^{n-1}(2^n + 1).$$

b) Input : $n = 2$
Output : `[[1, 1], [0, 0], [0, 2], [0, 3], [2, 0],...]`
Input : $n = 1$
Output : `[[0], [2], [3]]`

Trong lập luận (a−1) ta cần lập các xâu tứ phân độ dài n, xem Bài tập 1.17.

```python
def quaternary_strs(n):
    if n == 1:
        return [[i] for i in range(4)]
    S = []
    for i in range(4):
        for s in quaternary_strs(n-1):
            s = [i] + s
            S.append(s)
    return S

quaternary_strs(2)
```

Sau đó mới viết chương trình chính

```python
def quaternary_strs_1s_even(n):
    if n == 1:
        return [[0], [2], [3]]
    S = []
    for s in quaternary_strs(n-1):
        if s not in quaternary_strs_1s_even(n-1):
            s = [1] + s
            S.append(s)
```

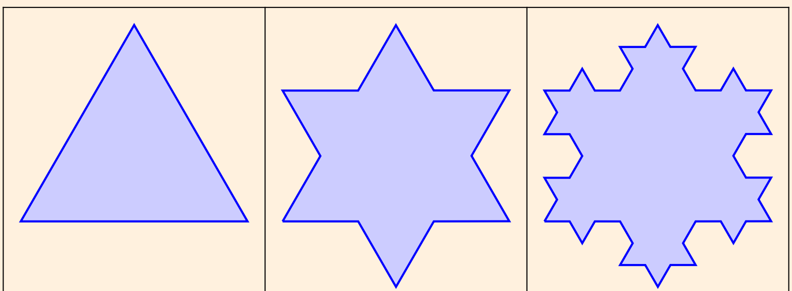
```
    for i in [0, 2, 3]:
        for s in quaternary_strs_1s_even(n-1):
            s = [i] + s
            S.append(s)
    return S

quaternary_strs_1s_even(2)
```

□

Tiếp theo, một ví dụ về hình học Fractal.

Ví dụ 9.10 ("Bông tuyết" Koch). [*]Cho với tam giác đều cạnh bằng 1, như phần (a) của hình dưới, được biến đổi thành sao David trong phần (b) bằng cách chia mỗi cạnh thành ba đoạn bằng nhau, bỏ đi đoạn ở giữa và gắn một tam giác đều mới, về phía ngoài, tại cạnh vừa bỏ đi. Tiếp tục quá trình này, ta biến đổi sao David thành đa giác ở phần (c).



(a) $n = 0$ (b) $n = 1$ (c) $n = 2$

Với $n \geq 0$, đặt s_n là diện tích của đa giác thứ n thu được từ tam giác đều ban đầu sau n phép biến đổi được mô tả ở trên.
Lập hệ thức đệ quy cho và giải s_n.

Giải. Ta có

$$\begin{aligned} s_n &= s_{n-1} + \text{ tổng diện tích các tam giác gắn thêm} \\ &= s_{n-1} + \text{ số tam giác gắn thêm} \times \text{ diện tích mỗi tam giác gắn thêm} \end{aligned}$$

Đặt e_n, ℓ_n là số cạnh, độ dài một cạnh của đa giác thứ n. Khi đó

$$e_n = 4e_{n-1}, \text{ với } e_0 = 3, \text{ và}$$

[*]Bông tuyết Koch được đưa ra năm 1904, bởi nhà toán học Thụy Điển Helge von Koch (1870 – 1924), với diện tích hữu hạn nhưng chu vi vô hạn.

$$\ell_n = \frac{\ell_{n-1}}{3}, \text{ với } \ell_0 = 1.$$

Suy ra

$$e_n = 3 \cdot 4^n, \quad \ell_n = \frac{1}{3^n}.$$

Nhắc lại, diện tích tam giác đều cạnh a là $\frac{\sqrt{3}}{4}a^2$. Suy ra

$$s_n = s_{n-1} + e_{n-1} \times \frac{\sqrt{3}}{4}\ell_n^2 = s_{n-1} + 3 \cdot 4^{n-1}\frac{\sqrt{3}}{4}\left(\frac{1}{3^n}\right)^2 = s_{n-1} + \frac{3\sqrt{3}}{16}\left(\frac{4}{9}\right)^n.$$

Giải hệ thức đệ quy với $s_0 = \frac{\sqrt{3}}{4}$, ta được

$$s_n = \frac{2\sqrt{3}}{5} - \frac{3\sqrt{3}}{20}\left(\frac{4}{9}\right)^n.$$

□

Trong quá trình giải, để đạt được nghiệm "đẹp", trong mã Python, cần thay $4/9$ bởi `Rational(4, 9)`

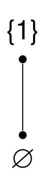
Ví dụ 9.11. Với $n \geq 1$, xét $X_n = \{1, 2, \dots, n\}$, tập lũy thừa $\mathcal{P}(X_n)$ của X_n, và quan hệ thứ tự $(\mathcal{P}(X_n), \subseteq)$ cùng với biểu đồ Hasse của nó.

a) Xây dựng biểu đồ Hasse của $(\mathcal{P}(X_n), \subseteq)$.

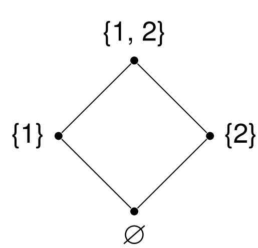
b) Đặt a_n là số cạnh trong biểu đồ Hasse của $(\mathcal{P}(X_n), \subseteq)$. Lập hệ thức đệ quy và giải a_n

Giải. a)

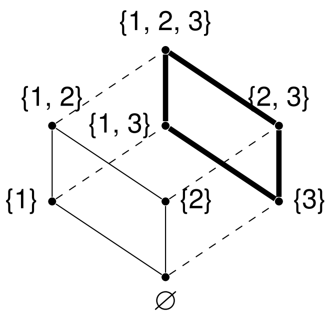
$n = 1$:



$n = 2$:



$n = 3$:



Đường nét liền: các đường trong biểu đồ Hasse của $(\mathcal{P}(X_2), \subseteq)$

Đường nét đậm: các đường liên quan đến các tập chứa số 3

Đường nét đứt: các đường nối các tập của $\mathcal{P}(X_2)$ với tập nào đó chứa số 3.

b) Từ mô tả cho trường hợp $n = 3$, ta chia $\mathcal{P}(X_n)$ thành hai phần $\mathcal{P}(X_{n-1})$ gồm các tập không chứa n và $\overline{\mathcal{P}} = \mathcal{P}(X_n) - \mathcal{P}(X_{n-1})$ gồm các tập chứa n. Khi đó các cạnh trong biểu đồ Hasse của $(\mathcal{P}(X_n), \subseteq)$ gồm

1) Các cạnh trong biểu đồ Hasse của $(\mathcal{P}(X_{n-1}), \subseteq)$, có a_{n-1} cạnh.

2) Các cạnh trong biểu đồ Hasse của $\overline{\mathcal{P}}$. Vì có cạnh nối $S, T \in \overline{\mathcal{P}}$ khi và chỉ khi có cạnh nối $S - \{n\}, T - \{n\} \in \mathcal{P}(X_{n-1})$, nên số cạnh trong trường hợp này bằng số cạnh của $(\mathcal{P}(X_{n-1}), \subseteq)$, là a_{n-1} cạnh.

3) Các cạnh nối giữa các $S \in \mathcal{P}(X_{n-1})$ với các $T \in \overline{\mathcal{P}}$. Vì đây là biểu đồ Hasse nên hoặc T là tập "nhỏ nhất" chứa S, hoặc S là tập nhỏ nhất chứa T. Nhưng $n \notin S, n \in T$, nên T là tập nhỏ nhất chứa S. Dễ thấy $T = S \cup \{n\}$. Vậy số cạnh trong trường hợp này bằng số tập $S \in \mathcal{P}(X_{n-1})$, là số tập con của tập cỡ $n - 1$, và bằng 2^{n-1}.

Theo quy tắc cộng

$$a_n = a_{n-1} + a_{n-2} + 2^{n-1} = 2a_{n-1} + 2^{n-1}, \quad n \geq 2,$$

trong đó $a_1 = 1$. Giải hệ thức đệ quy được

$$a_n = n \cdot 2^{n-1}.$$

□

Bài tập 9.3

9.1. Cho $a \in \mathbb{R}$, $n \in \mathbb{N}$. Từ định nghĩa đệ quy $a^n = a \cdot a^{n-1}$, ta xây dựng được thuật toán tính lũy thừa

```
def luy_thua(a, n):
    if n == 0:
        return 1
```

```
    return a * luy_thua(a, n-1)

luy_thua(2, 10)   # → 2^10 = 1024
```

Đặt a_n là số phép toán số học và so sánh trong thuật toán. Lập hệ thức đệ quy cho dãy (a_n) và tìm công thức tường minh của a_n.

9.2. Tìm quan hệ đệ quy của các cấp số nhân sau

a) $2, 10, 50, 250, \ldots$ b) $6, -18, 54, -162, \ldots$ c) $7, \frac{14}{5}, \frac{28}{25}, \frac{56}{125}, \ldots$

9.3. Giải các hệ thức đệ quy

a) $a_{n+1} - 1.5a_n = 0$

b) $4a_n - 5a_{n-1} = 0$

c) $3a_{n+1} - 4a_n = 0$, $a_1 = 5$

d) $2a_n - 3a_{n-1} = 0$, $a_4 = 81$

9.4. Cho quan hệ đệ quy $a_{n+1} - da_n = 0$. Biết $a_3 = \frac{153}{49}$ và $a_5 = \frac{1377}{240}$, tìm d.

9.5. Giải cấp số cộng (9.8) và cấp số nhân eqrefeq:cap-so-nhan bằng Python.

```
d, a0 = symbols('d a0')
rsolve(-a(n) + a(n-1) + d, a(n), {a(0): a0}).simplify()
```

```
q, a0 = symbols('q a0')
rsolve(-a(n) + q*a(n-1), a(n), {a(0): a0})
```

9.4 Hệ thức đệ quy tuyến tính thuần nhất hệ số hằng

Đây là dạng đặc biệt của (9.3) khi $f_n \equiv 0$

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}, \quad n \geq k. \tag{9.12}$$

Ở đây chỉ trình bày phương pháp giải tổng quát để hiểu được cấu trúc của nghiệm a_n. Với các hệ thức cụ thể, ta chủ yếu giải bằng Python.

Xét phương trình đặc trưng

$$\begin{aligned} & x^k = c_1 x^{k-1} + c_2 x^{k-2} + \cdots + c_{k-1} x + c_k \\ \Leftrightarrow\ & P(x) = x^k - c_1 x^{k-1} - c_2 x^{k-2} - \cdots - c_{k-1} x - c_k = 0, \end{aligned} \tag{9.13}$$

trong đó $P(x)$ gọi là đa thức đặc trưng của hệ thức đệ quy.

Phân tích $P(x)$ được các thừa số dạng sau:

1) $(x-a)^m$, $m \geq 1$, a gọi là nghiệm bội m của $P(x)$. Lập số hạng

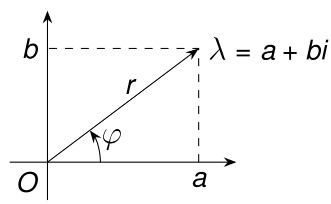
$$a^n P_{m-1}(n), \tag{9.14}$$

trong đó $P_{m-1}(n)$ là đa thức biến n bậc $m-1$ với hệ số bất định

$$P_{m-1}(n) = C_0 + C_1 n + C_2 n^2 + \cdots + C_{m-1} n^{m-1}. \tag{9.15}$$

2) $\left(x^2 + \alpha x + \beta\right)^m$, $m \geq 1$, trong đó $\alpha^2 - 4\beta < 0$, tức là tam thức bậc hai trong dấu ngoặc không có nghiệm thực. Khi đó $P(x)$ có nghiệm phức bội m, và ta chuyển nghiệm này sang dạng lượng giác

$$\begin{aligned} &\lambda = a \pm bi = r\left(\cos\varphi \pm i\sin\varphi\right) = e^{i\varphi} \\ \Leftrightarrow\ & r = \sqrt{a^2+b^2} \text{ và } \cos\varphi = \frac{a}{r}, \sin\varphi = \frac{b}{r}. \end{aligned} \tag{9.16}$$



Lập số hạng

$$r^n\left[P_{m-1}(n)\cos n\varphi + Q_{m-1}(n)\sin n\varphi\right] \tag{9.17}$$

trong đó $P_{m-1}(n)$ và $Q_{m-1}(n)$ cũng là các đa thức biến n bậc $m-1$ với hệ số bất định. Các hệ số bất định này thường được đánh chỉ số liên tiếp, và không được lặp lại trong các số hạng.

Nghiệm tổng quát của hệ thức đệ quy (9.12) là tổng của các số hạng có dạng (9.14) và (9.17) vừa lập ở trên, tương ứng với các nghiệm của phương trình đặc trưng.

Để tìm các hệ số bất định trong nghiệm tổng quát, cần k điều kiện ban đầu của dãy. Khi đó dẫn đến hệ phương trình tuyến tính với các ẩn là các biến bất định đó.

Ví dụ 9.12. Giải hệ thức $a_n = 3a_{n-2} - 4a_{n-3}$, $n \geq 3$. Xác định a_n biết $a_0 = 5$, $a_1 = -1$, $a_2 = 2$.

Giải. **Cách 1:** Giải hệ thức $a_n = 3a_{n-2} - 4a_{n-3}$ được $a_n = (-2)^n C_1 + C_0 + C_2 n$.

```python
from sympy import *

n = symbols('n')
a = symbols('a', cls=Function)
rsolve(
    -a(n) + 3*a(n-2) - 2*a(n-3),
    a(n)
)
```

Giải hệ thức với điều kiện $a_0 = 5$, $a_1 = -1$, $a_2 = 2$, được $a_n = (-2)^n - 3n + 4$.

```python
rsolve(
    -a(n) + 3*a(n-2) - 2*a(n-3),
    a(n),
    {a(0): 5, a(1): -1, a(2): 2}
)
```

Cách 2: Phương trình đặc trưng của hệ thức

$$x^3 = 3x - 2 \Leftrightarrow P(x) = x^3 - 3x + 2 = 0 \Leftrightarrow (x-1)^2(x+2) = 0$$

$$\Leftrightarrow \left[\begin{array}{lll} x = 1 & (\text{bội } m = 2) & \longrightarrow \text{số hạng } 1^n P_{2-1}(n) = 1^n(C_1 + C_2 n) \\ x = -2 & (\text{bội } m = 1) & \longrightarrow \text{số hạng } (-2)^n P_{1-1}(n) = (-2)^n C_3 \end{array}\right.$$

Suy ra $a_n = C_1 + C_2 n + (-2)^n C_3$.

```python
from sympy import *

x = symbols('x')
P = x**3 - 3*x + 2
P.factor()

C1, C2, C3 = symbols('C1 C2 C3')
ans = C1 + C2*n + (-2)**n * C3
```

Với $a_0 = 5$, $a_1 = -1$, $a_2 = 2$, ta có hệ

$$\begin{cases} C_1 & & & + & C_3 & = & 5 \\ C_1 & + & C_2 & - & 2C_3 & = & -1 \\ C_1 & + & 2C_2 & + & 4C_3 & = & 2 \end{cases}$$

```python
[ans.subs(n, i) for i in [0, 1, 2]]
```

Giải hệ được $C_1 = 4, C_2 = -3, C_3 = 1$. Vậy $a_n = 4 - 3n + (-2)^n \cdot 1 = 4 - 3n + (-2)^n$.

```python
eqns = [ans.subs(n, i) - a for i, a in zip([0, 1,
    2], [5, -1, 2])]
solve(eqns)
```

$\square$

Ví dụ 9.13. Giải hệ thức $a_{n+1} - 2a_n + 2a_{n-1} = 0,\ n \geq 1$, trong đó $a_0 = 1, a_1 = 2$.

Giải. **Cách 1:** Lệnh giải a_n

```python
from sympy import *

n = symbols('n', integer=True)  # biến nguyên bất
    định n
a = symbols('a', cls=Function)
ans = rsolve(
    a(n+1) - 2*a(n) + 2*a(n-1),
    a(n),
    {a(0): 1, a(1): 2}
)
ans  # (1/2 + i/2)(1 - i)^n + (1/2 - i/2)(1 + i)^n
```

cho kết quả

$$a_n = \frac{1+i}{2}(1-i)^n + \frac{1-i}{2}(1+i)^n$$

là biểu thức phụ thuộc số phức, trong khi thực chất a_n là số thực, cụ thể là số nguyên. Để biến đổi u_n về dạng số thực, ta cần chuyển các cơ số $1 - i$ và $1 + i$ trong nó về dạng dạng lượng giác, bằng hàm

```python
polar = lambda z: abs(z) * E**(I*arg(z))
```

Sau đó lấy mã của u_n trong Python, đang lưu trong biến `sol`

```python
print(ans)
```

được

```
(1/2 + I/2)*(1 - I)**n + (1/2 - I/2)*(1 + I)**n
```

Lúc này chỉ cần đặt hàm `polar` vào trước các cơ số đó. Ngoài ra để tránh mã `1/2` hiển thị là 0.5, mà muốn là $\frac{1}{2}$, ta thay bằng `Rational(1, 2)`

```
ans = (Rational(1, 2) + I/2) * polar(1 - I)**n +
   (Rational(1, 2) - I/2) * polar(1 + I)**n
ans
```

được

$$a_n = \left(\sqrt{2}e^{-\frac{i\pi}{4}}\right)^n \left(\frac{1}{2} + \frac{i}{2}\right) + \left(\sqrt{2}e^{\frac{i\pi}{4}}\right)^n \left(\frac{1}{2} - \frac{i}{2}\right)$$

vẫn chứa số phức. Tuy nhiên, chỉ cần đặt lệnh lấy phần thực của biểu thức này của a_n

```
re(ans)    # a_n = Re(a_n) = 2^(n/2) sin(πn/4) + 2^(n/2) cos(πn/4)
```

là đạt được nghiệm hoàn toàn là số thực

$$a_n = 2^{\frac{n}{2}} \left(\sin\frac{n\pi}{4} + \cos\frac{n\pi}{4}\right).$$

Cách 2: Phương trình đặc trưng

$$P(x) = x^2 - 2x + 2 = 0 \Leftrightarrow x = 1 \pm i \quad (\text{bội } m = 1)$$

```
from sympy import *

x = symbols('x')
P = x**2 - 2*x + 2
solve(P)
```

```
[1 - I, 1 + I]
```

Đây là hai nghiệm phức liên hợp, ta chọn một nghiệm, chẳng hạn $x = 1 + i$, và chuyển sang sạng lượng giác

$$x = \sqrt{2}\left(\cos\frac{\pi}{4} + i\sin\frac{\pi}{4}\right).$$

```
x = 1 + I
r = abs(x)
phi = arg(x)
```

Khi đó

$$\begin{aligned} a_n &= \left(\sqrt{2}\right)^n \left(P_{m-1}(n)\cos\frac{n\pi}{4} + Q_{m-1}(n)\sin\frac{n\pi}{4}\right) \\ &= \left(\sqrt{2}\right)^n \left(C_1\cos\frac{n\pi}{4} + C_2\sin\frac{n\pi}{4}\right). \end{aligned}$$

```
8  C1, C2 = symbols('C1 C2')
9  ans = r**n * (C1 * cos(n*phi) + C2 * sin(n*phi))
```

Với $a_0 = 1$, $a_1 = 2$, ta có hệ

$$\begin{cases} C_1 & & = & 1 \\ C_1 & + \quad C_2 & = & 2 \end{cases}$$

```
10 [ans.subs(n, i).simplify() for i in [0, 1]]
```

Giải hệ này, được $C_1 = 1$, $C_2 = 1$.

```
11 solve([ans.subs(n, i).simplify() - a for i, a in
      zip([0, 1], [1, 2])])
```

Do đó $a_n = \left(\sqrt{2}\right)^n \left(\cos\frac{n\pi}{4} + \sin\frac{n\pi}{4}\right)$.

□

Lưu ý, trong cách 1, dòng lệnh 14 muốn hiển thị kết quả như mong đợi, thì trong dòng lệnh 2 khi khai báo biến n phải có tùy chọn `integer=True` để chỉ định n là biến nguyên.

Các ví dụ sau, nếu không có gì đặc biệt, sẽ không trình bày quá trình giải hệ thức đệ quy.

Ví dụ 9.14. Với $n \geq 1$, a_n là số tập con của $\{1, 2, 3, \dots, n\}$ không có các số nguyên liên tiếp.

a) Với $1 \leq n \leq 4$, liệt kê các tập con của $\{1, 2, 3, \dots, n\}$ không có các số nguyên liên tiếp.

b) Lập hệ thức đệ quy và giải a_n.

c) Viết chương trình liệt kê các tập con mô tả ở trên.

Giải. a)

n	Tập con thỏa mãn	a_n
1	$\varnothing, \{1\}$	2
2	$\varnothing, \{1\}, \{2\}$	3
3	$\varnothing, \{1\}, \{2\}, \{3\}, \{1,3\}$	5
4	$\{4\}, \{1,4\}, \{2,4\}$, và $\varnothing, \{1\}, \{2\}, \{3\}, \{1,3\}$	8

b) Giả sử, với $n \geq 3$, xét tập $A \subseteq \{1, 2, 3, \ldots, n\}$ không có các số nguyên liên tiếp. Có hai khả năng:

1) $n \in A$: khi đó $n - 1 \notin A$, và $A - \{n\} \subseteq \{1, 2, \ldots, n-2\}$ không có các số nguyên liên tiếp. Theo định nghĩa, có a_{n-2} tập như vậy.

2) $n \notin A$: lúc này $A \subseteq \{1, 2, \ldots, n-1\}$ không có các số nguyên liên tiếp, và có a_{n-1} tập như vậy.

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

$$a_n = \frac{5+3\sqrt{5}}{10}\Big(\frac{1+\sqrt{5}}{2}\Big)^n - \frac{3\sqrt{5}-5}{10}\Big(\frac{1-\sqrt{5}}{2}\Big)^n.$$

c) Input : $n = 3$
Output : `[ [], [1], [2], [3], [1, 3] ]`
Input : $n = 1$
Output : `[ [], [1] ]`

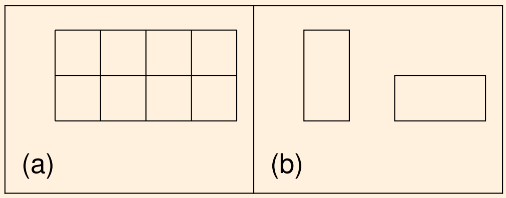
```python
def subsets_with_condition(n):
    if n == 1:
        return [[], [1]]
    if n == 2:
        return [[], [1], [2]]
    S = []
    for s in subsets_with_condition(n-1):
        S.append(s)
    for s in subsets_with_condition(n-2):
        s.append(n)
        S.append(s)
    return S

subsets_with_condition(3)
```

□

Trong ví dụ trên, ta thấy $a_n = F_{n+2} = \frac{1}{\sqrt{5}}\Big[\Big(\frac{1+\sqrt{5}}{2}\Big)^{n+2} - \Big(\frac{1-\sqrt{5}}{2}\Big)^{n+2}\Big]$.

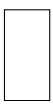
Ví dụ 9.15. Xét lưới cỡ $2 \times n$, với $n \in \mathbb{Z}^+$. Trường hợp $n = 4$ được cho trong phần (a) của hình dưới. Ta muốn phủ lưới này bằng các hình domino dọc cỡ 2×1 và ngang cỡ 1×2. Các domino này được cho trong phần (b) của hình.

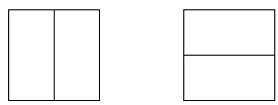


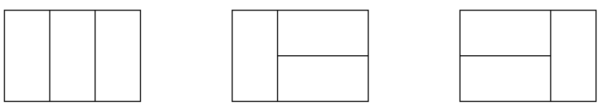
a) Liệt kê các cách phủ lưới ứng với $n = 1, 2, 3$.

b) Đặt a_n là số cách phủ lưới $2 \times n$. Lập hệ thức đệ quy và giải a_n.

Giải. a)

$n = 1$: 

$n = 2$: 

$n = 3$: 

b) Với $n \geq 3$, xét cột đầu tiên của lưới $2 \times n$. Có thể phủ cột này theo hai cách.

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

2) Bằng hai domino ngang để phủ cả hai cột đầu bên trái: phần còn lại, là lưới $2 \times (n-2)$, 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

$$\begin{aligned} a_n &= \Big(\frac{1}{2} - \frac{\sqrt{5}}{2}\Big)^n\Big(\frac{1}{2} - \frac{\sqrt{5}}{10}\Big) + \Big(\frac{1}{2} + \frac{\sqrt{5}}{2}\Big)^n\Big(\frac{\sqrt{5}}{10} + \frac{1}{2}\Big) \\ &= \frac{5+\sqrt{5}}{10}\left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{5-\sqrt{5}}{10}\left(\frac{1-\sqrt{5}}{2}\right)^n. \end{aligned}$$

□

Tương tự Ví dụ 9.14, trong ví dụ này

$$a_n = F_{n+1} = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}\right].$$

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.

a) Liệt kê các xâu như vậy ứng với $n = 1, 2, 3, 4$.

b) Lập hệ thức đệ quy và giải a_n.

c) Viết chương trình liệt kê các xâu như vậy độ dài n.

Giải. a)

$n = 1$: 0, 1

$n = 2$: 01, 10, 11

$n = 3$: **0**10 **0**11
101 **1**10 **1**11

$n = 4$: **0**101 **0**110 **0**111
1010 **1**011 **1**101 **1**110 **1**111

trong đó các xâu ứng với $n = 3, 4$ chia thành hai hàng, hàng trên gồm các xâu bắt đầu bằng 0.

b) 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, $a_n = a_{n-1} + a_{n-2}$, $n \geq 3$. trong đó $a_1 = 2$, $a_2 = 3$. Đây chính là dãy trong Ví dụ 9.14:

$$a_n = F_{n+2} = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+2}\right].$$

c) Input : $n = 3$
Outphut : `['010', '011', '101', '110', '111']`
Input : $n = 1$
Outphut : `['0', '1']`

```python
def binary_strs(n):
    if n == 1:
        return ['0', '1']
    if n == 2:
        return ['01', '10', '11']
    S = []
    for s in binary_strs(n-2):
        s = '01' + s
        S.append(s)
    for s in binary_strs(n-1):
        s = '1' + s
        S.append(s)
    return S

binary_strs(3)
```

□

Ví dụ 9.17. Trong một ngôn ngữ lập trình, xét các biểu thức số học hợp lệ, không có dấu ngoặc, gồm các chữ số 0, 1, 2,. . . , 9 và các phép toán hai ngôi $+, *, /$. Chẳng hạn, $3+4$ và $2+3*5$ là biểu thức số học hợp lệ; còn $8+*9$ thì không. Ở đây $2+3*5=17$, vì có thứ tự ưu tiên phép toán: với hai phép toán liên tiếp, phép nhân và chia (cùng cấp) thực hiện trước phép cộng, các phép toán cùng cấp thực hiện từ trái sang phải.
Với $n \in \mathbb{Z}^+$, đặt a_n là số biểu thức số học hợp lệ gồm n ký tự.

a) Xác định a_1, a_2.

b) Lập hệ thức đệ quy và giải a_n.

Giải. a) $a_1 = 10$, vì biểu thức hợp lệ có một ký tự sẽ là một trong 10 chữ số. Tiếp theo $a_2 = 100$, gồm các biểu thức 00, 01,. . . , 09, 10, 11,. . . , 99 (không xét trường hợp có dấu cộng ở đầu, chẳng hạn +5).

b) Khi $n \geq 3$, với mỗi biểu thức đếm bởi a_n, có hai khả năng:

1) $n-1$ ký tự đầu cũng là biểu thức hợp lệ, thì ký tự cuối phải là một chữ số. Số biểu thức như vậy là $10a_{n-1}$.

2) $n-1$ ký tự đầu không là biểu thức hợp lệ. Khi đó, ký tự thứ $n-1$ là phép toán, ký tự cuối là chữ số, và $n-2$ ký tự đầu là biểu thức hợp lệ. Vì

nhóm hai ký tự cuối có 29 lựa chọn, gồm +0, +1, ... , +9, $*0$, $*1$, ... , $*9$, và $/1$, $/2$, ... , $/9$, nên trường hợp này có $29a_{n-2}$ biểu thức.

Theo quy tắc cộng, $a_n = 10a_{n-1} + 29a_{n-2}$, $\forall n \geq 3$. Giải hệ thức đệ quy này với $a_1 = 10$ và $a_2 = 100$, được

$$a_n = \frac{5}{3\sqrt{6}}[(5 + 3\sqrt{6})^n - (5 - 3\sqrt{6})^n].$$

□

Trong Ví dụ 8.12, ta đã đếm các phân hạng đối xứng của n, tức là, số cách viết n thành dãy các số nguyên dương có tổng bằng n, mà cách đọc phép toán từ trái sang phải hay từ phải sang trái là như nhau. Đó là $a_n = 2^{\lfloor n/2 \rfloor}$.

Ví dụ 9.18. a) Liệt kê các phân hạng đối xứng của $n = 3, 4, 5, 6$.

b) Lập hệ thức đệ quy và giải a_n, là số phân hạng đối xứng của số nguyên dương n.

c) Viết chương trình liệt kê các phân hạng đối xứng của số nguyên dương n.

Giải. a)

$n = 3$	$n = 4$	$n = 5$	$n = 6$
(1) 3	(1) 4	(1) 5	(1) 6
(2) 1 + 1 + 1	(2) 2 + 2	(2) 2 + 1 + 2	(2) 3 + 3
	(3) 1 + 2 + 1	(3) 1 + 3 + 1	(3) 2 + 2 + 2
	(4) 1 + 1 + 1 + 1	(4) 1 + 1 + 1 + 1 + 1	(4) 2 + 1 + 1 + 2
			(5) 1 + 4 + 1
			(6) 1 + 2 + 2 + 1
			(7) 1 + 1 + 2 + 1 + 1
			(8) 1 + 1 + 1 + 1 + 1 + 1

b) Với mỗi phân hạng đối xứng của n, có hai khả năng:

1) Số đầu và số cuối lớn hơn 1. Khi đó, nếu trừ hai số này đi 1, ta được phân hạng đối xứng của $n - 2$. Số phân hạng như vậy là a_{n-2}.

2) Số đầu và số cuối bằng 1. Khi đó, nếu bỏ đi hai số này, ta lại được phân hạng đối xứng của $n - 2$. Số phân hạng như vậy là a_{n-2}.

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

$$a_n = \left(\frac{1}{2} + \frac{\sqrt{2}}{4}\right)(\sqrt{2})^n + \left(\frac{1}{2} - \frac{\sqrt{2}}{4}\right)(-\sqrt{2})^n.$$

```
from sympy import *

n = symbols('n')
a = symbols('a', cls=Function)
ans = rsolve(
    a(n) - 2*a(n-2),
    a(n),
    {a(1): 1, a(2): 2}
)
ans
```

Để rút gọn biểu thức của a_n, xét hai trường hợp $n = 2k$ hoặc $n = 2k + 1$. Khi đó, ta đều có $k = \lfloor n/2 \rfloor$, và $a_n = 2^k = 2^{\lfloor n/2 \rfloor}$.

```
k = symbols('k', integer=True)

ans.subs(n, 2*k).simplify()
ans.subs(n, 2*k+1).simplify()
```

c) Input : $n = 4$
Output : `[[4], [2, 2], [1, 2, 1], [1, 1, 1, 1]]`
Input : $n = 1$
Output : `[[1]]`

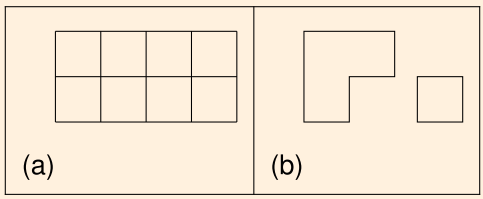
```
def symmetric_summands(n):
    if n == 1:
        return [[1]]
    if n == 2:
        return [[2], [1, 1]]
    S = []
    for s in symmetric_summands(n-2):
        s[0] += 1
        s[-1] += 1
        S.append(s)
```

```
    for s in symmetric_summands(n-2):
        s = [1] + s + [1]
        S.append(s)
    return S

symmetric_summands(4)
```

□

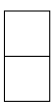
Ví dụ 9.19. Với $n \geq 1$, xét các cách phủ lưới $2 \times n$ như phần (a) trong hình dưới bằng hai loại gạch trong phần (b), gồm một viên cỡ 1×1 gọi là viên nhỏ, và viên kia gọi là viên lớn.

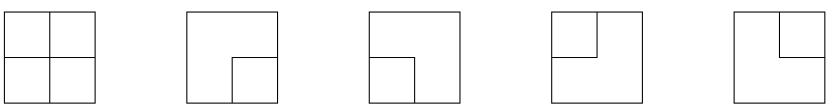


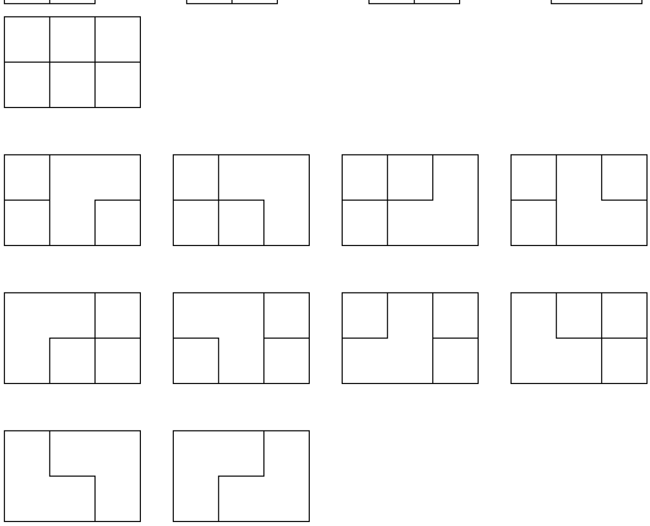
a) Liệt kê các cách phủ lưới $2 \times n$ ứng với $n = 1, 2, 3$.

b) Đặt a_n là số cách phủ lưới $2 \times n$. Lập hệ thức đệ quy và giải a_n.

Giải. a)

$n = 1$: 

$n = 2$: 

$n = 3$: 

b) Với mỗi cách phủ lưới $2 \times n$, xét ba khả năng phủ cột 1:

1) Cột 1 được phủ bằng hai viên 1×1, thì các cột còn lại, tức là lưới $2 \times (n-1)$, có a_{n-1} cách phủ.

2) Cột 1 chỉ có một viên 1×1, thì khi phủ tiếp bằng viên lớn, sẽ phủ hết hai cột đầu. Có hai cách phủ như vậy cho hai cột đầu. Sau đó, $n-2$ cột còn lại có a_{n-2} cách phủ, nên trường hợp này có $2a_{n-2}$ cách phủ.

3) Cột 1 không có viên 1×1 nào. Xét hai khả năng phủ cột 2:

   i) Cột 2 có một viên 1×1, thì có hai cột đầu được phủ hết, và bằng hai cách. Sau đó, $n-2$ cột còn lại có a_{n-2} cách phủ, nên trường hợp này có $2a_{n-2}$ cách phủ.

   ii) Cột 2 được phủ tiếp bằng viên lớn, thì ba cột đầu được phủ hết, và bằng hai cách. Sau đó, $n-3$ cột còn lại có a_{n-3} cách phủ, nên trường hợp này có $2a_{n-3}$ cách phủ.

Theo quy tắc cộng, $a_n = a_{n-1} + 2a_{n-2} + 2a_{n-2} + 2a_{n-3} = a_{n-1} + 4a_{n-2} + 2a_{n-3}$, $\forall n \geq 4$, trong đó $a_1 = 1, a_2 = 5, a_3 = 11$. Giải hệ thức đệ quy, được

$$a_n = (-1)^n - \frac{1}{\sqrt{3}}[(1+\sqrt{3})^n - (1-\sqrt{3})^n].$$

□

Ví dụ 9.20. Cho $a \in \mathbb{R}^*$, xét định thức cấp n

$$D_n = \begin{vmatrix} a & a & 0 & 0 & \dots & 0 & 0 & 0 \\ a & a & a & 0 & \dots & 0 & 0 & 0 \\ 0 & a & a & a & \dots & 0 & 0 & 0 \\ 0 & 0 & a & a & \dots & 0 & 0 & 0 \\ & & \dots & \dots & \dots & \dots & & \\ 0 & 0 & 0 & 0 & \dots & a & a & 0 \\ 0 & 0 & 0 & 0 & \dots & a & a & a \\ 0 & 0 & 0 & 0 & \dots & 0 & a & a \end{vmatrix}_n$$

Tìm công thức của D_n chỉ phụ thuộc n.

Giải. Khai triển D_n theo hàng đầu

$$D_n = a\underbrace{\begin{vmatrix} a & a & 0 & 0 & \dots & 0 & 0 & 0 \\ a & a & a & 0 & \dots & 0 & 0 & 0 \\ 0 & a & a & a & \dots & 0 & 0 & 0 \\ 0 & 0 & a & a & \dots & 0 & 0 & 0 \\ & & & \dots & \dots & & & \\ 0 & 0 & 0 & 0 & \dots & a & a & 0 \\ 0 & 0 & 0 & 0 & \dots & a & a & a \\ 0 & 0 & 0 & 0 & \dots & 0 & a & a \end{vmatrix}_{n-1}}_{D_{n-1}} - a\begin{vmatrix} a & a & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & a & a & 0 & \dots & 0 & 0 & 0 \\ 0 & a & a & a & \dots & 0 & 0 & 0 \\ 0 & 0 & a & a & \dots & 0 & 0 & 0 \\ & & & \dots & \dots & & & \\ 0 & 0 & 0 & 0 & \dots & a & a & 0 \\ 0 & 0 & 0 & 0 & \dots & a & a & a \\ 0 & 0 & 0 & 0 & \dots & 0 & a & a \end{vmatrix}_{n-1}$$

Khai triển định thức thứ hai theo cột đầu, được

$$D_n = aD_{n-1} - a(aD_{n-2}) = aD_{n-1} - a^2 D_{n-2}, \ \forall n \geq 3, \tag{1}$$

trong đó

$$D_1 = \left|[a]\right| = a, \quad D_2 = \begin{vmatrix} a & a \\ a & a \end{vmatrix} = 0. \tag{2}$$

```python
from sympy import *

a = symbols('a')
D1 = Matrix([[a]]).det()
D2 = Matrix([[a, a], [a, a]]).det()
```

Giải hệ thức đệ quy, được

$$D_n = \left(\frac{a\left(1-\sqrt{3}i\right)}{2}\right)^n \left(\frac{1}{2} + \frac{\sqrt{3}i}{6}\right) + \left(\frac{a\left(1+\sqrt{3}i\right)}{2}\right)^n \left(-\frac{3i}{3\sqrt{3}-3i} + \frac{\sqrt{3}}{3\sqrt{3}-3i}\right). \tag{3}$$

```python
n = symbols('n', integer=True)
ans = rsolve(
    -D(n) + a*D(n-1) - a**2 * D(n-2),
    D(n),
    {D(1): D1, D(2): D2}
)
```

Tương tự Ví dụ 9.13, ta xây dựng hàm chuyển các số phức về dạng lượng giác

```python
polar = lambda z: abs(z) * E**(I*arg(z))
```

Tiếp theo lấy mã Python của biểu thức của a_n

```
print(ans)
```

```
(a*(1 - sqrt(3)*I)/2)**n*(1/2 + sqrt(3)*I/6) + (a*(1
  + sqrt(3)*I)/2)**n*(-3*I/(3*sqrt(3) - 3*I) + sqrt
  (3)/(3*sqrt(3) - 3*I))
```

Nhập mã này vào dòng lệnh, lưu ý a_n có thừa số a^n, nên theo tính chất của lũy thừa có thể tạm thời loại thừa số a khỏi các cơ số, sau đó đặt các cơ số mới làm đối số của hàm `polar`. Ngoài ra, mã `1/2` cũng thay bởi `Rational(1, 2)`

```
ans = polar((1 - sqrt(3)*I)/2)**n * (Rational(1, 2) +
    sqrt(3)*I/6) + polar((1 + sqrt(3)*I)/2)**n * (-3*
   I/(3*sqrt(3) - 3*I) + sqrt(3)/(3*sqrt(3) - 3*I))
```

được

$$\frac{D_n}{a^n} = \left(-\frac{3i}{3\sqrt{3}-3i} + \frac{\sqrt{3}}{3\sqrt{3}-3i}\right) e^{\frac{i\pi n}{3}} + \left(\frac{1}{2} + \frac{\sqrt{3}i}{6}\right) e^{-\frac{i\pi n}{3}} \tag{4}$$

Cuối cùng, vì biểu thức đang xử lý là số thực, nên ta lấy phần thực của nó

```
re(ans).simplify()
```

thì được $\dfrac{D_n}{a^n} = \dfrac{2\sqrt{3}\sin\left(\pi\left(\frac{n}{3}+\frac{1}{3}\right)\right)}{3}$. Do đó

$$D_n = \frac{2}{\sqrt{3}} a^n \sin\frac{(n+1)\pi}{3}. \tag{5}$$

$\square$

Một cách khác để giải D_n. Xét phương trình đặc trưng của hệ thức (1)

$$x^2 = ax - a^2 \Leftrightarrow x^2 - ax + a^2 = 0 \Leftrightarrow x = \frac{a\left(1 \pm \sqrt{3}i\right)}{2} \quad \text{(bội } m = 1\text{)} \tag{6}$$

```
x = symbols('x')
sol = solve(x**2 - a*x + a**2, x)
```

```
[a*(1 - sqrt(3)*I)/2, a*(1 + sqrt(3)*I)/2]
```

Để hiển thị dạng toán học của các nghiệm này

```
for x in sol:
    display(x)
```

Ta chọn một nghiệm, chẳng hạn $x = \dfrac{a\left(1+\sqrt{3}i\right)}{2}$. Khử a ở vế phải để được số phức hằng số $\dfrac{x}{a} = \dfrac{1}{2} + \dfrac{\sqrt{3}i}{2} = x_0$, rồi đưa x_0 về dạng lượng giác

$$x_0 = r\left(\cos\varphi + i\sin\varphi\right) = 1\left(\cos\frac{\pi}{3} + i\sin\frac{\pi}{3}\right) \Rightarrow x = ax_0 = a\left(\cos\frac{\pi}{3} + i\sin\frac{\pi}{3}\right) \quad (7)$$

```
x = sol[1]
x0 = x / a

r = abs(x0)
phi = arg(x0)
```

Khi đó

$$D_n = a^n\left[P_{m-1}(n)\cos n\varphi + Q_{m-1}(n)\cos n\varphi\right] = a^n\left(C_1\cos\frac{n\pi}{3} + C_2\sin\frac{n\pi}{3}\right), \quad (8)$$

ở đây $P_{m-1}(n)$ và $Q_{m-1}(n)$ là các đa thức bậc 0, tức là hằng số. Với $D_1 = a$, $D_2 = 0$, ta có hệ

$$\begin{cases} a\left(\frac{C_1}{2} + \frac{\sqrt{3}C_2}{2}\right) = a \\ a^2\left(-\frac{C_1}{2} + \frac{\sqrt{3}C_2}{2}\right) = 0. \end{cases} \quad (9)$$

```
C1, C2 = symbols('C1 C2')
ans = a**n * (C1 * cos(n*phi) + C2 * sin(n*phi))
for i in [1, 2]:
    display(ans.subs(n, i))
```

Giải hệ này, được $C_1 = 1$, $C_2 = \dfrac{\sqrt{3}}{3}$. Thay vào D_n và rút gọn

$$D_n = a^n\left(\frac{\sqrt{3}\sin\left(\frac{\pi n}{3}\right)}{3} + \cos\left(\frac{\pi n}{3}\right)\right) = \frac{2\sqrt{3}a^n\sin\left(\pi\left(\frac{n}{3}+\frac{1}{3}\right)\right)}{3} = \frac{2}{\sqrt{3}}a^n\sin\frac{(n+1)\pi}{3}. \quad (10)$$

```
sol = solve(ans.subs(n, i) - D for i, D in zip([1,
    2], [a, 0]))
ans.subs(sol[0]).simplify()
```

Bài tập 9.4

9.6. Giải các hệ thức đệ quy

a) $a_n = 5a_{n-1} + 6a_{n-2}$; $a_0 = 1$, $a_1 = 3$

b) $2a_{n+2} - 11a_{n+1} + 5a_n = 0$; $a_0 = 2$, $a_1 = -8$

c) $a_{n+2} + a_n = 0$; $a_0 = 0$, $a_1 = 3$

d) $a_n - 6a_{n-1} + 9a_{n-2} = 0$; $a_0 = 5$, $a_1 = 12$

e) $a_n + 2a_{n-1} + 2a_{n-1} = 0$; $a_0 = 1$, $a_1 = 3$

9.7. Xét quan hệ đệ quy $a_{n+2} + ba_{n+1} + ca_n = 0$. Biết $a_0 = 0$, $a_1 = 1$, $a_2 = 4$, và $a_3 = 37$, tìm b, c và giải hệ thức đệ quy.

9.8. Giải hệ thức đệ quy $a_{n+2} = a_{n+1}a_n$; $a_0 = 1$, $a_1 = 2$.

9.9. Tìm và giải hệ thức đệ quy của $D_n = \begin{vmatrix} 2 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & \dots & 0 & 0 & 0 \\ & & & \dots\dots & \dots\dots & & & \\ 0 & 0 & 0 & 0 & \dots & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 2 \end{vmatrix}_n$

9.10. Giải hệ thức đệ quy $a_{n+2}^2 - 5a_{n+1}^2 + 4a_n^2 = 0$; $a_0 = 4$, $a_1 = 13$.

9.5 Hệ thức đệ quy tuyến tính không thuần nhất hệ số hằng

Hệ thức đã được định nghĩa ở Phần 9.1

$$a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k} + f_n,\ \ n \geq k, \tag{9.3}$$

Giả sử

a) $\overline{a}_n$ là nghiệm tổng quát của phương trình thuần nhất tương ứng

$$a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k},\ \ n \geq k. \tag{9.12}$$

b) a_n^* là một nghiệm riêng của (9.3)

Khi đó nghiệm tổng quát của (9.3) là

$$a_n = \overline{a}_n + a_n^*. \tag{9.18}$$

Một số trường hợp có thể tìm được a_n^*, dựa vào dạng của f_n:

1)

$$f_n = a^n P_d(n) \tag{9.19}$$

trong đó $x = a \in \mathbb{R}$ là nghiệm bội m của đa thức đặc trưng (9.13), $P_d(n)$ là đa thức biến n bậc d. Khi đó

$$a_n^* = a^n n^m P_d^*(n) \tag{9.20}$$

trong đó $P_d^*(n)$ là đa thức bậc d với hệ số bất định.

2)

$$f_n = r^n \left[P_{d_1}(n) \cos n\varphi + Q_{d_2}(n) \sin n\varphi\right] \tag{9.21}$$

trong đó $P_{d_1}(n)$, $Q_{d_2}(n)$ là các đa thức bậc d_1, d_2, và $x = r\left(\cos\varphi + i\sin\varphi\right)$ là nghiệm bội m của đa thức đặc trưng. Đặt $d = \max\{d_1, d_2\}$, khi đó

$$a_n^* = r^n n^m \left[P_d^*(n) \cos n\varphi + Q_d^*(n) \sin n\varphi\right] \tag{9.22}$$

trong đó $P_d^*(n)$, $Q_d^*(n)$ là các đa thức bậc d với hệ số bất định.

Các hệ số của $P_d^*(n)$, $Q_d^*(n)$ thường được tìm bằng cách thay a_n^* vào (9.3).

Lưu ý, nếu $x = a$ hoặc $x = r\left(\cos\varphi + i\sin\varphi\right)$ không là nghiệm của đa thức đặc trưng, thì gọi là nghiệm bội $m = 0$.

Ngoài ra, kết quả sau bổ sung một phương pháp hiệu quả để tìm nghiệm riêng, thường áp dụng khi f_n không có dạng (9.19) hay (9.21), nhưng có thể tách thành tổng của các dạng này.

Nguyên lý chồng chất nghiệm

Xét hệ thức đệ quy

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f_n^{(1)} + f_n^{(2)} \tag{9.23}$$

Giả sử $u_n^{(i)}$, $i = 1, 2$, là một nghiệm riêng của hệ thức

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f_n^{(i)} \tag{9.24}$$

Khi đó $u_n^* = u_n^{(1)} + u_n^{(2)}$ là một nghiệm riêng của (9.23).

Ví dụ 9.21. Cho đoạn mã định nghĩa dãy Fibonacci bằng đệ quy

```
1 def F(n):
2     if n == 0:
3         return 0
4     elif n == 1:
5         return 1
6     return F(n-1) + F(n-2)
```

Đặt a_n là số phép toán số học và phép so sánh dùng để tính $F(n)$. Lập hệ thức đệ quy và giải a_n.

Giải. Xét đoạn mã khi:

$n = 0$: dòng 2 thực hiện 1 phép so sánh, điều kiện thỏa mãn nên thoát chương trình tại dòng 3. Ta có $a_0 = 1$.

$n = 1$: dòng 2 thực hiện 1 phép so sánh, điều kiện không thỏa mãn thì thực hiện tiếp 1 phép so sánh ở dòng 4. Tại đây, điều kiện thỏa mãn nên thoát chương trình ở dòng 5. Ta có $a_1 = 2$.

$n > 1$: 2 phép so sánh ở dòng 2 và dòng 4 vẫn được thực hiện, và cả hai đều không thỏa mãn, nên chương trình thực hiện dòng 6, với

i) 2 phép trừ để tính các đối số $n-1$, $n-2$, và 1 phép cộng.

ii) a_{n-1} phép toán để tính $F(n-1)$, và a_{n-2} phép toán để tính $F(n-2)$.

Do đó, ta có hệ thức đệ quy

$$a_n = 2 + (3 + a_{n-1} + a_{n-2}) = a_{n-1} + a_{n-2} + 5.$$

Giải hệ thức ta được

$$\begin{aligned} a_n &= \left(\frac{1}{2} - \frac{\sqrt{5}}{2}\right)^n \left(3 - \frac{4\sqrt{5}}{5}\right) + \left(\frac{1}{2} + \frac{\sqrt{5}}{2}\right)^n \left(\frac{4\sqrt{5}}{5} + 3\right) - 5 \\ &= \left(3 + \frac{4}{\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(3 - \frac{4}{\sqrt{5}}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n - 5. \end{aligned}$$

```
from sympy import *

n = symbols('n')
a = symbols('a', cls=Function)
rsolve(
    -a(n) + a(n-1) + a(n-2) + 5,
    a(n),
    {a(0): 1, a(1): 2}
)
```

□

Bài tập 9.5

9.11. Giải các hệ thức đệ quy

a) $a_{n+1} - a_n = 2n + 3$, $a_0 = 1$

b) $a_{n+1} - a_n = 3n^2 - n$, $a_0 = 3$

c) $a_{n+1} - 2a_n = 5$, $a_0 = 1$

d) $a_{n+1} - 2a_n = 2^n$, $a_0 = 1$

9.12. Giải các hệ thức đệ quy

a) $a_{n+2} + 3a_{n+1} + 2a_n = 3^n$; $a_0 = 0$, $a_1 = 1$

b) $a_{n+2} + 4a_{n+1} + 4a_n = 7$; $a_0 = 1$, $a_1 = 2$

9.13. Giải hệ thức đệ quy $a_{n+2} - 6a_{n+1} + 9a_n = 3 \cdot 2^n + 7 \cdot 3^n$; $a_0 = 1$, $a_1 = 4$

9.14. Tìm nghiệm tổng quát của quan hệ đệ quy $a_{n+3} - 3a_{n+2} + 3a_{n+1} - a_n = 3 + 5n$.

9.15. Nghiệm tổng quát của quan hệ đệ quy $a_{n+2} + b_1a_{n+1} + b_2a_n = b_3n + b_4$, với các hằng số b_i, $1 \leq i \leq 4$, là $c_1 2^2 + c_2 3^n + n - 7$. Tìm b_i, $1 \leq i \leq 4$.

9.16. Giải các quan hệ đệ quy

a) $a_{n+2}^2 - 5a_{n+1}^2 + 6a_n^2 = 7n$, $a_0 = a_1 = 1$

b) $a_n^2 - 2a_{n-1} = 0$, $a_0 = 2$. [*Gợi ý:* đặt $b_n = \log_2 a_n$]

9.6 Phương pháp tính tổng

Giả sử cần tính tổng

$$a_n = \sum_{i=1}^{n} f_i = f_1 + f_2 + \cdots + f_n \tag{9.25}$$

trong đó f_n thường có dạng (9.19), (9.21), hoặc nếu không, là tổng của các dạng này.

Nhận xét

$$a_n = \left(f_1 + f_2 + \cdots + f_{n-1}\right) + f_n = a_{n-1} + f_n, \;\; n \geq 2. \tag{9.26}$$

Ta muốn hệ thức trên cũng đúng với $n = 1$, tức là định nghĩa thêm a_0 thỏa mãn $a_1 = a_0 + f_1$. Từ (9.25), $a_1 = f_1$, suy ra

$$a_0 = 0.$$

Đây là hệ thức đệ quy tuyến không thuần nhất cấp một hệ số hằng, có phương trình đặc trưng

$$x = 1 \quad (\text{bội } m = 1),$$

nên phương trình thuần nhất tương ứng có nghiệm tổng quát

$$\overline{a}_n = 1^n P_{m-1}(n) = C.$$

Giả sử a_n^* là một nghiệm riêng của (9.26), khi đó (9.26) có nghiệm tổng quát

$$a_n = \overline{u}_n + a_n^* = C + a_n^*$$

Vì $a_0 = 0$ nên $C + a_0^* = 0$, suy ra $C = -a_0^*$. Do đó

$$a_n = a_n^* - a_0^*. \tag{9.27}$$

9.7 Phương pháp hàm sinh

Giả sử dãy a_n có hệ thức đệ quy. Xét hàm sinh $f(x) = \sum_{n=0}^{\infty} a_n x^n$. Căn cứ vào hệ thức đệ quy của a_n, bằng phép biến đổi phù hợp, ta biểu diễn được $f(x)$ theo theo x và chính $f(x)$. Từ đó, ta giải được $f(x)$ là một hàm sơ cấp. Khi đó, a_n là hệ số của x^n trong khai triển Maclaurin của hàm này.

9.7.1 Hệ thức đệ quy tuyến tính hệ số hằng

Lại xét hệ thức

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f_n,\ \ n \geq k, \tag{9.3}$$

Biến đổi

$$\begin{aligned}
f(x) &= \sum_{n=0}^{\infty} a_n x^n = \sum_{n=0}^{k-1} a_n x^n + \sum_{n=k}^{\infty} a_n x^n \\
&= \sum_{n=0}^{k-1} a_n x^n + \sum_{n=k}^{\infty} \left(c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f_n\right) x^n \\
&= \sum_{n=0}^{k-1} a_n x^n + \sum_{n=k}^{\infty} c_1 a_{n-1} x^n + \sum_{n=k}^{\infty} c_2 a_{n-2} x^n + \cdots + \sum_{n=k}^{\infty} c_k a_{n-k} x^n + \sum_{n=k}^{\infty} x^n f_n \\
&= \sum_{n=0}^{k-1} a_n x^n + c_1 x \sum_{n=k}^{\infty} a_{n-1} x^{n-1} + c_2 x^2 \sum_{n=k}^{\infty} a_{n-2} x^{n-2} + \cdots + \\
&\qquad + c_k x^k \sum_{n=k}^{\infty} a_{n-k} x^{n-k} + \sum_{n=k}^{\infty} x^n f_n \\
&= \sum_{n=0}^{k-1} a_n x^n + c_1 x \left[f(x) - \sum_{n=0}^{k-2} a_n x^n \right] + c_2 x^2 \left[f(x) - \sum_{n=0}^{k-3} a_n x^n \right] + \cdots + \\
&\qquad + c_k x^k \left[f(x) \right] + \sum_{n=k}^{\infty} x^n f_n.
\end{aligned}$$

Gom các số hạng chứa $f(x)$

$$\begin{aligned}
&\left(c_k x^k + \cdots + c_2 x^2 + c_1 x - 1\right) f(x) \\
= &- \sum_{n=0}^{k-1} a_n x^n + c_1 x \sum_{n=0}^{k-2} a_n x^n + c_2 x^2 \sum_{n=0}^{k-3} a_n x^n + \cdots + c_{k-1} x^{k-1} a_0 - \sum_{n=k}^{\infty} x^n f_n
\end{aligned} \tag{9.28}$$

trong đó chuỗi $\sum_{n=k}^{\infty} x^n f_n$ có thể tính bằng cách

Cách 1: Tính tổng riêng $\sum_{n=k}^{m} x^n f_n$ theo phương pháp tương tự Phần 9.6. Sau đó lấy giới hạn của tổng riêng khi $m \to \infty$.

Cách 2: Nếu f_n có dạng (9.19), hay tổng các số hạng có dạng này, thì áp dụng phương pháp trong Ví dụ 8.14.

Tiếp theo, từ (9.28) ta giải được $f(x)$. Thông thường $f(x)$ có dạng phân thức, khi đó, ta đưa nó về dạng tổng các phân thức đơn giản, mà ta đã biết khai triển Maclaurin của các phân thức loại này.

Cuối cùng, nhắc lại, a_n là hệ số của x^n trong khai triển Maclaurin của $f(x)$.

9.7.2 Hệ các hệ thức đệ quy tuyến tính bậc nhất hệ số hằng

Cho các dãy $\left(a_n^{(i)}\right)$, $1 \leq i \leq k$, thỏa mãn hệ các hệ thức đệ quy

$$\begin{cases} a_{n+1}^{(i)} = c_{i1}a_n^{(1)} + c_{i2}a_n^{(2)} + \cdots + c_{ik}^{(k)}a_n^{(k)} + f_n^{(i)} \\ i = 1, 2, \dots, k, \end{cases} \tag{9.29}$$

trong đó $a_0^{(i)}$ cho trước, c_{ij}, $1 \leq i, j \leq k$ là các hằng số.

Xét các hàm sinh $f_i(x) = \sum\limits_{n=0}^{\infty} a_n^{(i)} x^n$. Biến đổi

$$\begin{aligned} f_i(x) &= \sum_{n=0}^{\infty} a_n^{(i)} x^n = a_0^{(i)} + \sum_{n=1}^{\infty} a_n^{(i)} x^n = a_0^{(i)} + \sum_{n=0}^{\infty} a_{n+1}^{(i)} x^{n+1} \\ &= a_0^{(i)} + \sum_{n=0}^{\infty} \left(c_{i1}a_n^{(1)} + c_{i2}a_n^{(2)} + \cdots + c_{ik}^{(k)}a_n^{(k)} + f_n^{(i)}\right) x^{n+1} \\ &= a_0^{(i)} + c_{i1}x\sum_{n=0}^{\infty} a_n^{(1)}x^n + c_{i2}x\sum_{n=0}^{\infty} a_n^{(2)}x^n + \cdots + c_{ik}x\sum_{n=0}^{\infty} a_n^{(k)}x^n + \sum_{n=0}^{\infty} x^{n+1}f_n^{(i)} \\ &= a_0^{(i)} + c_{i1}xf_1(x) + c_{i2}xf_2(x) + \cdots + c_{ik}xf_k(x) + \sum_{n=0}^{\infty} x^{n+1}f_n^{(i)} \end{aligned} \tag{9.30}$$

ta được hệ phương trình tuyến tính theo các ẩn $f_i(x)$, $1 \leq i \leq k$.

Các bước tiếp theo tiến hành tương tự Mục 9.7.1.

Nhận xét: Hệ thức đệ quy tuyến tính cấp k hệ số hằng (9.3) là trường hợp riêng của hệ k hệ thức đệ quy tuyến tính cấp một hệ số hằng (9.29), bằng cách đặt

$$a_n^{(1)} = a_{n-1}, \quad a_n^{(2)} = a_{n-2}, \quad a_n^{(3)} = a_{n-3}, \dots \quad a_n^{(k)} = a_{n-k},$$

và khi đó

$$\begin{cases} a_{n+1}^{(1)} = a_n = c_1a_n^{(1)} + c_2a_n^{(2)} + \cdots + c_ka_n^{(k)} + f_n \\ a_{n+1}^{(2)} = a_{n-1} = a_n^{(1)} \\ a_{n+1}^{(3)} = a_{n-2} = a_n^{(2)} \\ \cdots \\ a_{n+1}^{(k)} = a_{n-k+1} = a_n^{(k-1)} \end{cases}$$

với điều kiện ban đầu

$$a_k^{(1)} = a_{k-1}, \quad a_k^{(2)} = a_{k-2}, \quad a_k^{(3)} = a_{k-3}, \ldots \quad a_k^{(k)} = a_0$$

Bảng sau liệt các khả năng giải các loại hệ thức trên của các công cụ tính toán

Loại hệ thức \ Công cụ	MATLAB	Mathematica	Python
(9.3)		✓	✓
(9.29)		✓	

Ví dụ 9.22. Giải hệ các hệ thức đệ quy

$$\begin{cases} a_{n+1} = 5a_n - 3b_n - 2^n(3n+5) \\ b_{n+1} = 6b_n - 4b_n + 2n + 1 \end{cases} \quad \text{với} \begin{cases} a_0 = 5 \\ b_0 = 4 \end{cases}$$

Giải. Xét các hàm sinh $f(x) = \sum_{n=0}^{\infty} a_n x^n$ và $g(x) = \sum_{n=0}^{\infty} b_n x^n$ ứng với các dãy (a_n) và (b_n). Biến đổi

$$\begin{aligned} f(x) &= a_0 + \sum_{n=1}^{\infty} a_n x^n = 5 + \sum_{n=0}^{\infty} a_{n+1} x^{n+1} = 5 + \sum_{n=0}^{\infty} \left[5a_n - 3b_n - 2^n(3n+5)\right] x^{n+1} \\ &= 5 + \sum_{n=0}^{\infty} 5a_n x^{n+1} - \sum_{n=0}^{\infty} 3b_n x^{n+1} - \sum_{n=0}^{\infty} 2^n(3n+5) x^{n+1} \\ &= 5 + 5x \cdot f(x) - 3x \cdot g(x) - \frac{x(5-4x)}{(2x-1)^2} \\ \Rightarrow & \ (5x-1) f(x) - 3x \cdot g(x) + 5 - \frac{x(5-4x)}{(2x-1)^2} = 0 \end{aligned} \tag{1}$$

```python
from sympy import *

x, n = symbols('x n')
Sum(2**n * (3*n + 5) * x**(n+1), (n, 0, oo)).doit().
    simplify()
```

$$\begin{cases} \frac{x(5-4x)}{(2x-1)^2} & \text{for } x > -\frac{1}{2} \wedge x < \frac{1}{2} \\ \sum_{n=0}^{\infty} 2^n x^{n+1} \cdot (3n+5) & \text{otherwise} \end{cases}$$

và tương tự

$$g(x) = b_0 + \sum_{n=1}^{\infty} b_n x^n = 4 + \sum_{n=0}^{\infty} b_{n+1} x^{n+1} = 4 + \sum_{n=0}^{\infty} (6a_n - 4b_n + 2n + 1) x^{n+1}$$

$$= 4 + \sum_{n=0}^{\infty} 6a_n x^{n+1} - \sum_{n=0}^{\infty} 4b_n x^{n+1} + \sum_{n=0}^{\infty} (2n+1) x^{n+1}$$

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

$$\Rightarrow 6x \cdot f(x) - (4x+1) g(x) + 4 + \frac{x(x+1)}{(x-1)^2} = 0 \tag{2}$$

```
Sum((2*n + 1) * x**(n+1), (n, 0, oo)).doit().simplify
    ()
```

$$\begin{cases} \frac{x(x+1)}{(x-1)^2} & \text{for } x > -1 \wedge x < 1 \\ \sum_{n=0}^{\infty} x^{n+1} \cdot (2n+1) & \text{otherwise} \end{cases}$$

Giải (1) và (2) được

$$f(x) = \frac{-36x^5 + 124x^4 - 96x^3 - 8x^2 + 27x - 5}{8x^6 - 20x^5 + 10x^4 + 13x^3 - 17x^2 + 7x - 1}$$

$$g(x) = \frac{-44x^5 + 178x^4 - 181x^3 + 46x^2 + 13x - 4}{8x^6 - 20x^5 + 10x^4 + 13x^3 - 17x^2 + 7x - 1}$$

```
f, g = symbols('f g')
eq1 = (5*x - 1) * f - 3*x * g + 5 - x*(5 - 4*x)/(2*x
    - 1)**2
eq2 = 6*x * f - (4*x + 1) * g + 4 + x*(x + 1)/(x - 1)
    **2
ans = solve([eq1, eq2], [f, g])
ans      # có kiểu từ điển
ans[f]
```

```
{f: (-36*x**5 + 124*x**4 - 96*x**3 - 8*x**2 + 27*x - 5)
    /(8*x**6 - 20*x**5 + 10*x**4 + 13*x**3 - 17*x**2 + 7*x
    - 1),
 g: ... }
```

Khai triển $f(x)$ thành tổng các phân thức đơn giản

$$\begin{aligned}
f(x) &= -\frac{5}{2x-1} + \frac{2}{(2x-1)^2} + \frac{3}{(2x-1)^3} - \frac{2}{x+1} + \frac{3}{(x-1)^2} \\
&= 5(1-2x)^{-1} + 2(1-2x)^{-2} - 3(1-2x)^{-3} - 2(1+x)^{-1} + 3(1-x)^{-2}.
\end{aligned}$$

```
ans[f].apart()
```

Khi đó a_n là hệ số của x^n trong khai triển Maclaurin của $f(x)$

$$\begin{aligned}
a_n &= 5\binom{-1}{n}(-2)^n + 2\binom{-2}{n}(-2)^n - 3\binom{-3}{n}(-2)^n - 2\binom{-1}{n} + 3\binom{-2}{n}(-1)^n \\
&= 5\binom{1+n-1}{n}2^n + 2\binom{2+n-1}{n}2^n - 3\binom{3+n-1}{n}2^n \\
&\qquad - 2\binom{1+n-1}{n}(-1)^n + 3\binom{2+n-1}{n} \\
&= 2^n\left[5\binom{n}{n} + 2\binom{n+1}{n} - 3\binom{n+2}{n}\right] - 2\binom{n}{n}(-1)^n + 3\binom{n+1}{n} \\
&= 2^n\left[5\cdot 1 + 2(n+1) - 3\frac{(n+2)(n+1)}{2}\right] - 2\cdot 1(-1)^n + 3(n+1) \\
&= 2^{n-1}\left(-3n^2 - 5n + 8\right) - 2(-1)^n + 3n + 3.
\end{aligned}$$

Tương tự

$$\begin{aligned}
g(x) &= -\frac{5}{2x-1} + \frac{3}{(2x-1)^2} + \frac{3}{(2x-1)^3} - \frac{4}{x+1} + \frac{1}{x-1} + \frac{4}{(x-1)^2} \\
&= 5(1-2x)^{-1} + 3(1-2x)^{-2} - 3(1-2x)^{-3} - 4(1+x)^{-1} - (1-x)^{-1} + \\
&\qquad + 4(1-x)^{-2},
\end{aligned}$$

và

$$\begin{aligned}
b_n &= 5\binom{-1}{n}(-2)^n + 3\binom{-2}{n}(-2)^n - 3\binom{-3}{n}(-2)^n - 4\binom{-1}{n} \\
&\qquad - \binom{-1}{n}(-1)^n + 4\binom{-2}{n}(-1)^n \\
&= 5\binom{1+n-1}{n}2^n + 3\binom{2+n-1}{n}2^n - 3\binom{3+n-1}{n}2^n - 4\binom{1+n-1}{n}(-1)^n \\
&\qquad - \binom{1+n-1}{n} + 4\binom{2+n-1}{n} \\
&= 2^n\left[5\binom{n}{n} + 3\binom{n+1}{n} - 3\binom{n+2}{n}\right] - 4\binom{n}{n}(-1)^n - \binom{n}{n} + 4\binom{n+1}{n}
\end{aligned}$$

$$= 2^n\left[5\cdot 1+3\left(n+1\right)-3\frac{\left(n+2\right)\left(n+1\right)}{2}\right]-4\cdot 1(-1)^n-1+4\left(n+1\right)$$
$$= 2^{n-1}\left(-3n^2-3n+10\right)-4(-1)^n+4n+3.$$

□

Phương pháp hàm sinh cũng có thể giải một số hệ thức đệ quy đặc biệt, chẳng hạn, hệ thức đệ quy tuyến tính hai biến.

Ví dụ 9.23. Với $n, r \in \mathbb{N}$, đặt $c(n, r)$ là số cách chọn r vật, có lặp, từ n vật. Chứng minh

a) $c(n,r) = c(n-1,r) + c(n,r-1),\ \forall n, r \geq 1.$

b) $c(n,r) = \binom{n+r-1}{r}.$

Giải. a) [*] Xét x là một vật trong n vật. Trong r vật chọn ra, chỉ có hai khả năng:

1) Vật x không được chọn. Khi đó r vật được chọn từ $n-1$ kia. Ta có $c(n-1,r)$ cách.

2) Vật x được chọn ít nhất một lần. Khi đó ta cần chọn $r-1$ vật có lặp từ n vật đó, rồi chọn tiếp một vật x. Ta có $c(n,r-1)$ cách.

Theo quy tắc cộng, $c(n,r) = c(n-1,r) + c(n,r-1),\ \forall n, r \geq 1$. Ngoài ra $c(n,0) = 1,\ \forall n \geq 0$ và $c(0,r) = 0,\ \forall r > 0$.

b) Đặt $f_n(x) = \sum_{r=0}^{\infty} c(n,r)x^r$ là hàm sinh dãy $c(n,0), c(n,1), c(n,2), \ldots$ Ta có

$$\begin{aligned} f_n(x) &= c(n,0) + \sum_{r=1}^{\infty} c(n,r)x^r = 1 + \sum_{r=1}^{\infty}[c(n-1,r) + c(n,r-1)]x^r \\ &= 1 + \sum_{r=1}^{\infty} c(n-1,r)x^r + \sum_{r=1}^{\infty} c(n,r-1)x^r \\ &= f_{n-1}(x) + x\sum_{r=1}^{\infty} c(n,r-1)x^{r-1} = f_{n-1}(x) + xf_n(x) \end{aligned}$$

[*] Lập luận này đã được trình bày trong Ví dụ 1.26(c)

Suy ra

$$f_n(x) - x\, f_n(x) = f_{n-1}(x) \Rightarrow f_n(x)\cdot(1-x) = f_{n-1}(x) \Rightarrow f_n(x) = \frac{f_{n-1}(x)}{1-x}.$$

Vì $f_0 = c(0,0) + c(0,1)x + c(0,2)x^2 + \cdots = 1 + 0 + 0 + \cdots = 1$, nên

$$f_n = \frac{1}{(1-x)^n} = (1-x)^{-n}.$$

Do đó $c(n,r)$ là hệ số của x^r trong $(1-x)^{-n}$, đó là

$$c(n,r) = \binom{-n}{r}(-1)^r = \binom{n+r-1}{r}.$$

□

Bài tập 9.7

9.17. Giải các quan hệ đệ quy bằng phương pháp hàm sinh

a) $a_{n+1} - a_n = 3^n$, $a_0 = 1$

b) $a_{n+1} - a_n = n^2$, $a_0 = 1$

c) $a_{n+2} - 3a_{n+1} + 2a_n = 0$; $a_0 = 1$, $a_1 = 6$

d) $a_{n+2} - 2a_{n+1} + a_n = 2^n$; $a_0 = 1$, $a_1 = 2$

9.18. Cho n vật phân biệt. Đặt $a(n,r)$ là số cách chọn, không lặp, r vật từ n vật, với $0 \le r \le n$. Ở đây $a(n,r) = 0$ nếu $r > n$. Dùng quan hệ đệ quy $a(n,r) = a(n-1,r-1) + a(n-1,r)$, với $n, r \ge 1$ để chứng minh $f(x) = (1+x)^n$ sinh dãy $a(n,r)$, $r \ge 0$.

9.19. Giải hệ các hệ thức đệ quy

a) $a_{n+1} = -2a_n - 4b_n$, $b_{n+1} = 4a_n + 6b_n$; $a_0 = 1$, $b_0 = 0$

b) $a_{n+1} = 2a_n - b_n + 2$, $b_{n+1} = -a_n + 2b_n - 1$; $a_0 = 0$, $b_0 = 1$

9.8 Hệ thức đệ quy phi tuyến đặc biệt

Ví dụ 9.24. Xét bài toán liệt kê các cách tính tích $a_0 a_1 a_2 \cdots a_n$, $n \ge 1$, bằng cách đặt các dấu ngoặc ( ).

a) Liệt kê các cách tính ứng với $n = 1, 2, 3, 4$.

b) Đặt C_n là số cách tính $a_0 a_1 \cdots a_n$. Lập hệ thức đệ quy cho dãy (C_n).

c) Tìm công thức tường minh của C_n.

Giải. a)

$n = 1$: $(A_0)\,(A_1)$

$n = 2$: $(A_0)\,((A_1)\,(A_2))$ $\quad ((A_0)\,(A_1))\,(A_2)$

$n = 3$: $(((A_0)\,(A_1))\,(A_2))\,(A_3)$ $\quad ((A_0)\,((A_1)\,(A_2)))\,(A_3)$

$((A_0)\,(A_1))\,((A_2)\,(A_3))$

$(A_0)\,(((A_1)\,(A_2))\,(A_3))$

$(A_0)\,((A_1)\,((A_2)\,(A_3)))$

b) Xét các cách đặt dấu ngoặc cho phép tính cuối cùng, tức là sau a_k, với $k = \overline{0, n-1}$:

$$\underbrace{(a_0 a_1 a_2 \cdots a_k)}_{c_k \text{ cách}}\underbrace{(a_{k+1} a_{k+2} \cdots a_n)}_{c_{n-k-1} \text{ cách}}$$

Áp dụng quy tắc đếm cộng và nhân, ta có

$$c_n = \sum_{k=0}^{n-1} c_k c_{n-k-1} \tag{9.31}$$

trong đó để phù hợp với quy tắc nhân, ta cần $c_0 = 1$.

c) 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

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

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

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,

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

□

Ví dụ 9.25.

Ví dụ 9.26.

Ví dụ 9.27.

9.9 Thuật toán chia để trị

Để giải một bài toán với đầu vào cỡ n:

1) Chia bài toán thành a bài toán nhỏ cùng dạng với bài toán ban đầu, nhưng cỡ là $\frac{n}{b}$, chính xác là $\left\lfloor \frac{n}{b} \right\rfloor$ hoặc $\left\lceil \frac{n}{b} \right\rceil$. Ở đây $a \geq 1, b \geq 2$ là các hằng số nguyên.

2) Tổng hợp kết của từ các bài toán nhỏ, để suy ra kết quả của bài toán ban đầu.

Gọi $f(n)$ là độ phức tạp của thuật toán với cỡ đầu vào n, $h(n)$ là độ phức tạp của bước (2). Khi đó

$$f(n) = af\left(\frac{n}{b}\right) + h(n).$$

Để đơn giản ta xét các giá trị của n là $1, b, b^2, b^3, ...$, tức là $n = b^k$ với $k \in \mathbb{N}$. Khi đó

$$f(b^k) = af(b^{k-1}) + h(b^k).$$

Đặt $u_k = f(b^k)$, ta được hệ thức đệ quy tuyến tính không thuần nhất cấp một hệ số hằng

$$u_k = au_{k-1} + h(b^k).$$

trong đó $u_0 = f(1)$. Khi đó

$$f(n) = f(b^k) = u_k = u_{\log_b n}.$$

Một số thuật toán chia để trị điển hình như tìm số lớn nhất của dãy, tìm kiếm nhị phân, và sắp xếp trộn.

Ví dụ 9.28. Trình bày một thuật toán chia để trị để tìm số lớn nhất M của dãy $a_1, a_2, \ldots, a_n$, và đánh giá độ phức tạp của thuật toán.

Giải. Xét thuật toán chia để trị sau:

1) Nếu dãy chỉ có một phần tử, thì $M = a_1$. Nếu không thì xuống thực hiện bước (2).

2) Chia đôi dãy được hai dãy con $a_1, a_2, \dots, a_k$ và $a_{k+1}, a_{k+2}, \dots, a_n$. Mỗi dãy có $\frac{n}{2}$ số. Gọi M_1, M_2 là số lớn nhất của hai dãy con này.

3) $M = \max\{M_1, M_2\}$.

Chia đôi dãy $a_1, a_2, \dots, a_n$ thành hai dãy con.

```
1  def my_max(a):
2    n = len(a)            # 1 phép tính, 1 phép gán
3    if n == 1:            # 1 phép so sánh
4      return a[0]
5    k = n // 2            # 1 phép toán
6    M1 = my_max(a[:k])    # bài toán với cỡ đầu vào n/2, và 1 phép gán
7    M2 = my_max(a[k:])    # như dòng 6
8    if M1 > M2:           # 1 phép so sánh
9      return M1
10   else:
11     return M2

12 my_max([2, 8, 6, 1, 0, 5, 4, 1, 7])  # → 8
```

Gọi $f(n)$ là số phép so sánh để tìm số lớn nhất của dãy cỡ n, được dùng để đánh giá độ phức tạp của thuật toán. Ta có

$$f(n) = \underbrace{1}_{\text{bước 1}} + \underbrace{2 \cdot f\left(\frac{n}{2}\right)}_{\text{bước 2}} + \underbrace{1}_{\text{bước 3}} = 2f\left(\frac{n}{2}\right) + 2.$$

Xét $n = 2^k$, $a_k = f(n) = f(2^k)$ thì

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

trong đó $a_0 = f(1) = 0$.

Giải hệ thức đệ quy, ta được $a_k = 3 \cdot 2^k - 2$, hay $f(n) = 3n - 2 \in O(n)$. Thuật toán có độ phức tạp tuyến tính.

Thay vì chỉ đếm số phép so sánh, nếu đặt $f(n)$ là bao gồm các phép toán, phép so sánh, và phép gán của thuật toán trên, thì $f(n) = 2f\left(\frac{n}{2}\right) + 7$, tương ứng $a_k = 2a_{k-1} + 7$, với $a_0 = f(1) = 3$. Khi đó $a_k = 5 \cdot 2^k - 2$, tức là $f(n) = 5n - 2$. □

Ví dụ 9.29 (Tìm kiếm nhị phân)**.** Tìm vị trí của x trong dãy đơn điệu $a_1 < a_2 < \ldots < a_n$: nêu một thuật toán chia để trị và đánh giá độ phức tạp của thuật toán.

Giải. Xét thuật toán chia để trị tìm vị trí của x trong dãy $a_i, a_{i+1}, \ldots, a_j$ với $i \leq j$. Ban đầu $i = 1, j = n$.

1) Nếu dãy chỉ có một phần tử, tức là $i = j$, thì chỉ cần so sánh x với a_i [Nếu $x = a_i$ thì i là vị trí của x trong dãy, ngược lại không tìm được x trong dãy]. Ngược lại thì xuống thực hiện bước (2).

2) Chia đôi dãy được hai dãy con $a_i, a_{i+1}, \ldots, a_k$ và $a_{k+1}, a_{k+2}, \ldots, a_j$. Mỗi dãy có $\frac{n}{2}$ số.
Do tính đơn điệu của dãy, nên nếu $x \leq a_k$ thì chỉ cần tìm x trong dãy thứ nhất, và nếu ngược lại, chỉ cần tìm x trong dãy thứ hai. Tức là, chỉ thực hiện một bài bài toán với cỡ đầu vào $\frac{n}{2}$.

```
1  def binary_search(x, a, i, j):
2    if i == j:            # 1 phép so sánh
3      if x == a[i]:       # 1 phép so sánh
4        return i
5      else:
6        return 'Không thấy'
7    k = (i + j) // 2      # 2 phép toán, 1 phép gán
8    if x <= a[k]:         # 1 phép so sánh
9      return binary_search(x, a, i, k)       # bài toán
         với cỡ đầu vào n/2
10   else:
11     return binary_search(x, a, k + 1, j)   # như dòng 9

12 a = [0, 2, 3, 6, 7, 10, 11, 15]
13 binary_search(10, a, 0, len(a) - 1)        # → 5
```

Đặt $f(n)$ là số phép so sánh cho bài toán với đầu vào cỡ n. Khi đó

$$f(n) = \underbrace{2}_{\text{dòng 2, 8}} + \underbrace{f\left(\frac{n}{2}\right)}_{\text{dòng 9 hoặc 11}}$$

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 9.30 (Sắp xếp trộn)**.** Sắp xếp dãy $a_1, a_2, \dots, a_n$ 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 $a_1, a_2, \dots, a_k$ và $a_{k+1}, a_{k+2}, \dots, a_n$ 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*

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

thì với mọi $n = 1, b, b^2, \ldots$

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

```
rsolve(-u(k) + a*u(k-1) + c, u(k), {u(0): c}).
    simplify()
rsolve(-u(k) +   u(k-1) + c, u(k), {u(0): c}).
    simplify()
```

Với giả thiết của định lý, ta thường đánh giá

$$f(n) \in \begin{cases} O(\log_b n)) & \text{nếu } a = 1 \\ O(n^{\log_b a}) & \text{nếu } a \geq 2. \end{cases}$$

Hệ quả 9.1. *Cho* $a, b, c \in \mathbb{Z}^+$, $b \geq 2$, *và* $f : \mathbb{Z}^+ \to \mathbb{R}^+ \cup \{0\}$. *Nếu*

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

thì với mọi $n = 1, b, b^2, \ldots$

$$f(n) \in \begin{cases} O(\log_b n)) & \text{nếu } a = 1 \\ O(n^{\log_b a}) & \text{nếu } a \geq 2. \end{cases}$$

Bài tập 9.9

9.20. Cho $f : \mathbb{Z}^+ \to \mathbb{R}$. Giải quan hệ $f(n)$ trên tập S đã cho, và xác định dạng O–lớn của f trên S.

a) $f(1) = 5$; $f(n) = 4f\left(\frac{n}{3}\right) + 5$, $n = 3, 9, 27, \ldots$; $S = \{3^i \mid i \in \mathbb{N}\}$

b) $f(1) = 7$; $f(n) = f\left(\frac{n}{5}\right) + 7$, $n = 5, 25, 125, \ldots$; $S = \{5^i \mid i \in \mathbb{N}\}$

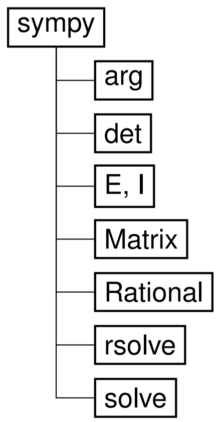
c) $f(1) = 0$; $f(n) = 2f\left(\frac{n}{5}\right) + 3$, $n = 5, 25, 125, \ldots$; $S = \{5^i \mid i \in \mathbb{N}\}$

d) $f(1) = 1$; $f(n) = f\left(\frac{n}{2}\right) + 2$, $n = 2, 4, 8, \ldots$; $S = \{2^i \mid i \in \mathbb{N}\}$

9.21. Cho $a, b, c \in \mathbb{Z}^+$, $b \geq 2$, và $d \in \mathbb{N}$. Giải hệ thức đệ quy

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

Tóm tắt lệnh Python



Bài tập bổ sung

Tài liệu tham khảo

[1] NumPy community. *NumPy User Guide*. phiên bản 1.22.4. 535 trang. URL: https://numpy.org/doc/stable.

[2] Judi J. McDonald David C. Lay Steven R. Lay. *Linear Algebra and Its Applications*. In lần thứ 6. Pearson, 2022. 755 trang.

[3] Ralph P. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. In lần thứ 5. Pearson Addison-Wesley, 2004. 992 trang.

[4] Ralph P. Grimaldi. *Discrete and Combinatorial Mathematics: Instructor's Solutions Manual*. In lần thứ 5. Pearson Addison-Wesley, 2004. 465 trang.

[5] Thomas Koshy. *Catalan Numbers with Applications*. Oxford University Pres, 2009. 439 trang.

[6] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. In lần thứ 8. McGraw-Hill Education, 2019. 1118 trang.

[7] Edward R. Scheinerman. *Mathematics: A Discrete Introduction*. In lần thứ 3. Brooks/Cole, Cengage Learning, 2013. 506 trang.

[8] Watson S. Stewart J. Clegg D. *Calculus: Early Transcendentals*. In lần thứ 9. Cengage Learning, 2011. 1421 trang.

[9] SymPy Development Team. *SymPy Documentation*. phiên bản 1.8. 2750 trang. URL: https://github.com/sympy/sympy/releases.