Mục lục

Phần I

Cơ sở của Toán rời rạc

Chương 1

Nguyên lý đếm cơ bản

1.1 Quy tắc cộng, nhân

Quy tắc cộng: Nếu có hai phương án thực hiện một công việc:

a) phương án một có m cách thực hiện, phương án hai có n cách thực hiện; và

b) không thể thực hiện đồng thời hai phương án,

thì có $m + n$ cách thực hiện công việc.

Quy tắc cộng có thể mở rộng với nhiều phương án hơn, miễn là không có cặp phương án nào thực hiện được đồng thời.

Quy tắc nhân: Nếu có hai bước thực hiện một công việc:

a) bước một có m cách thực hiện, và

b) ứng với mỗi kết quả của bước một, bước hai có n cách thực hiện,

thì có $m \times n$ cách thực hiện công việc.

Quy tắc nhân cũng có thể được mở rộng đối với công việc phân thành nhiều giai đoạn.

Đôi khi cần kết hợp nhiều nguyên tắc đếm để giải một bài toán. Trường hợp điển hình, một công việc gồm nhiều bước, mà số cách thực hiện ở bước sau thay đổi phụ thuộc kết quả của bước trước.

Ví dụ 1.1. Trong thư viện của trường đại học, giáo trình về khoa học máy tính gồm 40 giáo trình của nhà xuất bản A, và 50 giáo trình của nhà xuất bản B.

a) Có bao nhiêu cách để sinh viên chọn một giáo trình về khoa học máy tính?

b) Có bao nhiêu cách để sinh viên chọn hai giáo trình về khoa học máy tính, trong đó mỗi nhà xuất bản được chọn một cuốn?

Giải. a) Theo quy tắc cộng, số cách chọn một giáo trình về khoa học máy tính là

$$40 + 50 = 90.$$

b) Để chọn hai giáo trình, mỗi nhà xuất bản được chọn một cuốn:

Bước 1: Chọn một giáo trình của nhà xuất bản A, có 40 cách.

Bước 2: Ứng với giáo trình đã chọn của nhà xuất bản A, chọn tiếp một giáo trình của nhà xuất bản B, có 50 cách.

Theo quy tắc nhân, số cách chọn hai giáo trình như vậy là

$$40 \times 50 = 2\,000.$$

□

Ví dụ 1.2. Trong môn học ngôn ngữ lập trình, giảng viên giới thiệu ba ngôn ngữ Python, Java, và C++. Mỗi ngôn ngữ có 5 sách tham khảo.
Có bao nhiêu cách để sinh viên chọn một sách trong số đó để học?

Giải. Ví dụ này có thể giải bằng quy tắc cộng hoặc quy tắc nhân.

Cách 1: Theo quy tắc cộng, số cách sinh viên chọn một cuốn sách là

$$5 + 5 + 5 = 15.$$

Cách 2: Để chọn một cuốn sách, sinh viên thực hiện hai bước

Bước 1: Chọn một ngôn ngữ lập trình, có 3 cách.

Bước 2: Ứng với ngôn ngữ lập trình đó, chọn một sách tham khảo, có 5 cách.

Theo quy tắc nhân, số cách chọn một cuốn sách là

$$3 \times 5 = 15.$$

□

Ví dụ 1.3. Câu lạc bộ kịch của trường đại học tổ chức buổi thử vai cho một vở diễn. Có 6 nam và 8 nữ thử vai nam và nữ chính.
Đạo diễn có bao nhiêu cách chọn cặp diễn viên chính?

Giải. Theo quy tắc nhân, số cách chọn cặp diễn viên chính là

$$\underset{\text{nam}}{\underset{\downarrow}{6}} \times \underset{\text{nữ}}{\underset{\downarrow}{8}} = 48$$

□

Ví dụ 1.4. Một biển số xe máy có dạng “29-D1 371.68”, trong đó 29 là mã vùng, D1 là mã quận gồm một chữ cái tiếng Anh viết hoa và một chữ số, và 371.68 là thứ tự đăng ký xe, là một xâu thập phân gồm năm chữ số. Nếu không xét đến các yếu tố khác, thì có thể tạo bao nhiêu biển số xe có dạng:

a) Mã vùng Hà Nội (từ 29 đến 33, và 40)?

b) Mã vùng 29, thứ tự đăng ký gồm các chữ số khác nhau?

Giải. a) Ta mô tả số cách chọn tương ứng với mỗi thành phần trên biển số

$$\begin{array}{ccccccccccc}
\boxed{29} & \text{-} & D & 1 & \quad & 3 & 7 & 1 & \cdot & 6 & 8 \\
\downarrow & & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\
6 & & 26 & 10 & & 10 & 10 & 10 & & 10 & 10
\end{array}$$

Theo quy tắc nhân, số các biển xe mã vùng Hà Nội là

$$6 \times 26 \times \underbrace{10 \times 10 \times \cdots \times 10}_{6} = 156 \times 10^6 = 156 \text{ triệu}.$$

```
6 * 26 * 10**6
```

b) Số cách chọn tương ứng với mỗi thành phần trên biển số

$$\begin{array}{ccccccccccc}
29 & \text{-} & D & 1 & \quad & 3 & 7 & 1 & \cdot & 6 & 8 \\
 & & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\
 & & 26 & 10 & & 10 & 9 & 8 & & 7 & 6
\end{array}$$

Theo quy tắc nhân, số các biển xe mã vùng 29 có thứ tự đăng ký gồm các chữ số khác nhau là

$$26 \times 10 \times 10 \times 9 \times 8 \times 7 \times 6 = 7\,862\,400.$$

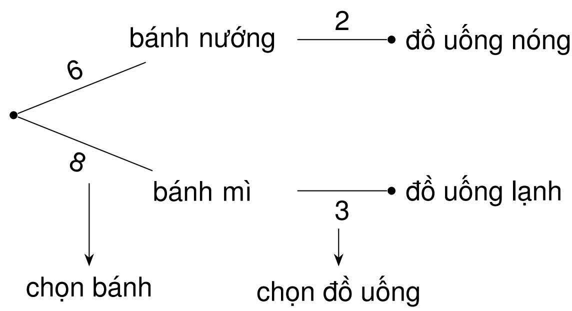
□

Ví dụ 1.5. Một cửa hàng phục vụ 6 loại bánh nướng, 8 loại bánh mì và 5 loại đồ uống (cà phê nóng, trà nóng, trà đá, cola và nước cam). Có bao nhiêu cách gọi thực đơn gồm bánh và đồ uống, trong đó bánh nướng phải đi kèm đồ uống nóng, hoặc bánh mì đi kèm đồ uống lạnh.

Giải. Theo quy tắc nhân, có $6 \times 2 = 12$ cách mua bánh nướng và đồ uống nóng; và có $8 \times 3 = 24$ cách đặt bánh mì và đồ uống lạnh.

Theo quy tắc cộng, có $12 + 24 = 36$ cách gọi thực đơn như trên. □

Có thể mô tả lập luận trên bằng biểu đồ cây



trong đó mỗi nút ghi kết quả của từng bước, và số cách để đạt kết quả đó ghi trên cạnh tương ứng.

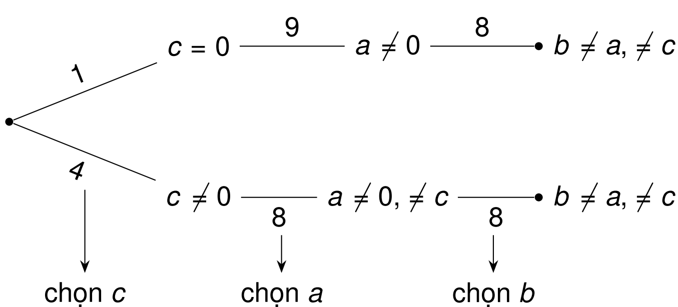
Ví dụ 1.6. Mô tả cách đếm các số tự nhiên chẵn có ba chữ số khác nhau.

Giải. Ta cần đếm các số có dạng $\overline{abc}$, với các chữ số $a \neq 0$, c chẵn, và a, b, c khác nhau.

Với các ràng buộc của a, b, c lần lượt là

a: $\neq 0, \neq b, \neq c$ $\qquad$ b: $\neq a, \neq c$, và $\qquad$ c: chẵn, $\neq a, \neq b$

ta sẽ lần lượt chọn giá trị cho c, a và b.



Kết hợp quy tắc cộng và nhân, ta đếm được

$$1 \cdot 9 \cdot 8 + 4 \cdot 8 \cdot 8 = 328 \text{ số.}$$

□

Các ví dụ từ Ví dụ 1.7 đến 1.10 về thuật toán, lập trình, và đánh giá độ phức tạp của thuật toán.

Ví dụ 1.7. Với số nguyên dương m, n, xét đoạn chương trình

```
for i = 1 to m do
```

```
2     for j = 1 to n do
3         print i, j
```

a) Viết chương trình bằng Python với $m = 3, n = 4$. Đếm số lệnh `print` đã thực hiện.

b) Với m, n bất kỳ, lệnh `print` thực hiện bao nhiêu lần?

Giải. a) Dùng biến `counter` đi cùng với lệnh `print` để đếm số lệnh này đã thực hiện sau mỗi chu trình của vòng lặp

```python
m, n = 4, 3   # gán đồng thời
counter = 0
for i in range(1, m+1):
    for j in range(1, n+1):
        counter += 1
        print(counter, i, j)
```

b) – Dòng 3 (i, j cố định): có 1 lệnh `print`
  – Dòng $2-3$ (i cố định): vì $j = \overline{1, n}$ nên có n lệnh `print` được thực hiện
  – Dòng $1-3$ tức toàn bộ chương trình: vì $i = \overline{1, m}$ nên theo quy tắc nhân, số lệnh `print` được thực hiện là $m \times n$.

□

Ví dụ 1.8. Với số nguyên dương n, xét đoạn chương trình

```
counter = 0
for i = 1 to n do
    for j = 1 to i do
        counter += 1
```

a) Viết đoạn chương trình bằng Python với $n = 4$. Cho biết giá trị của biến `counter` sau khi thực hiện xong chương trình.

b) Với n bất kỳ, tìm giá trị của biến `counter` sau khi thực hiện xong chương trình.

Giải. a)

```
1 n = 4
2 counter = 0
3 for i in range(1, n+1):
4     for j in range(1, i+1):
5         counter += 1

6 print(counter)
```

Sau khi chạy chương trình, biến `counter` nhận giá trị 10.

b) Trước khi bước vào vòng lặp, `counter` = 0, và mỗi chu trình của vòng lặp sẽ

- duyệt qua một cặp (i, j) với $i = 1, 2, \ldots, n$ và $j = 1, 2, \ldots, i$.
- `counter` tăng 1 đơn vị.

Vì vậy, giá trị sau cùng của `counter` bằng số cặp (i, j) được duyệt.

Với mỗi $i = \overline{1, n}$, dòng 3–4 duyệt i cặp (i, j). Theo quy tắc cộng, số cặp (i, j) được duyệt, tức giá trị của biến `counter` sau khi thực hiện chương trình, là

$$\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}. \qquad (*)$$

$\square$

Tính tổng trong $(*)$ bằng thư viện `sympy`

```
1 from sympy import *

2 n, i = symbols('n i')
3 Sum(i, (i, 1, n)).doit().simplify()
```

trong đó dòng 2 khai báo các biến bất định n, i có trong biểu thức tổng, dòng 3 gồm lệnh `Sum` để lập biểu thức tổng, lệnh `doit` để tính tổng đó, và lệnh `simplify` để rút gọn kết quả tính được. Ta nên gõ lần lượt từng lệnh để quan sát kết quả rồi mới nhập lệnh tiếp theo.

Ví dụ 1.9. Nhân ma trận $A = (a_{ij})_{m\times n}$ với $B = (b_{ij})_{n\times p}$, được ma trận $C = AB = (c_{ij})_{m\times p}$ xác định bởi $c_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}$. Giả mã của thuật toán như sau

```
for i = 1 to m do
```

```
    for j = 1 to p do
        c_ij = 0
        for k = 1 to n to
            c_ij = c_ij + a_ik × b_kj
```

a) Viết chương trình Python tính tích hai ma trận A và B. Minh họa với

$$A = \begin{bmatrix} 2 & 0 & -1 \\ 1 & 3 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & -1 & 1 & 0 \\ 2 & 3 & -1 & 4 \\ -3 & 0 & -2 & 1 \end{bmatrix}$$

b) Thuật toán thực hiện bao nhiêu phép toán số học?

Giải. a) $AB = \begin{bmatrix} 3 & -2 & 4 & -1 \\ 12 & 8 & 2 & 10 \end{bmatrix}$

```python
A = [[ 2,  0, -1],
     [ 1,  3, -2]]
B = [[ 0, -1,  1, 0],
     [ 2,  3, -1, 4],
     [-3,  0, -2, 1]]
```

Cách 1: Lập trình

```python
def matrix_mul(A, B):
    m, n = len(A), len(B)
    p = len(B[0])
    C = [[0 for j in range(p)] for i in range
  (m)]
    for i in range(m):
        for j in range(p):
            for k in range(n):
                C[i][j] += A[i][k] * B[k][j]
    return C

matrix_mul(A, B)
```

Cách 2: Dùng thư viện numpy

```
6 import numpy as np

7 np.dot(A, B)
```

b)
- Dòng 5 thực hiện 2 phép toán số học.
- Dòng 4 – 5, theo quy tắc nhân, thực hiện $n \times 2 = 2n$ phép toán.
- Dòng 2 – 5, cũng theo quy tắc nhân, thực hiện $p \times 2n = 2np$ phép toán.
- Dòng 1 – 5, tức toàn bộ chương trình, thực hiện $m \times 2np = 2mnp$ phép toán.

□

Ví dụ 1.10. Thuật toán *sắp xếp nổi bọt*:

```
1 def BubbleSort(x):              # x = [x_0, x_1, ..., x_{n-1}]
2     n = len(x)                  # độ dài/cỡ của x
3     for i in range(n-1):        # duyệt từ đầu, x_0, tới gần cuối, x_{n-2}.
4         for j in range(n-1,i,-1):   # duyệt từ cuối, x_{n-1} về kế sau x_i, tức x_{i+1}
5             if x[j] < x[j-1]:
6                 x[j-1], x[j] = x[j], x[j-1]   # đổi chỗ
7     return x   # trả về kết quả cho hàm

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

Thuật toán dùng bao nhiêu phép so sánh để sắp xếp dãy cỡ n?

Giải.

- Dòng 5, hay 5 – 6 cũng vậy, với $i = \overline{0, n-2}$, $j = \overline{n-1 \downarrow i+1}$: có 1 phép so sánh.
- Dòng 4 – 6, với $i = \overline{0, n-2}$: có $(n-1) - (i+1) + 1 = n - i - 1$ phép so sánh.
- Dòng 3 – 6, và do đó toàn bộ chương trình: theo quy tắc cộng, số phép so sánh là

$$\sum_{i=0}^{n-2} (n - i - 1) = (n-1) + (n-2) + \cdots + 2 + 1 = \frac{n(n-1)}{2}.$$

□

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

Bài tập 1.1

1.1. Tính

a) $\sum_{i=1}^{6}(i^2 + 1)$ b) $\sum_{j=-2}^{2}(j^3 - 1)$ c) $\sum_{i=0}^{10}[1 + (-1)^i]$ d) $\sum_{i=1}^{6} i(-1)^i$

e) $\sum_{k=n}^{2n}(-1)^k$, trong đó n là số nguyên dương lẻ.

1.2. Biểu diễn các biểu thức bằng ký hiệu tổng (hay Sigma). Trong ý (a), (d) và (e), n là số nguyên dương.

a) $\dfrac{1}{2!} + \dfrac{1}{3!} + \dfrac{1}{4!} + \cdots + \dfrac{1}{n!}$

b) $1 + 4 + 9 + 16 + 25 + 36 + 49$

c) $1^3 - 2^3 + 3^3 - 4^3 + 5^3 - 6^3 + 7^3$

d) $\dfrac{1}{n} + \dfrac{2}{n+1} + \dfrac{3}{n+2} + \cdots + \dfrac{n+1}{2n}$

e) $n - \dfrac{n+1}{2!} + \dfrac{n+2}{4!} - \dfrac{n+3}{6!} + \cdots + (-1)^n \dfrac{2n}{(2n)!}$

1.3. a) Có bao nhiêu cách để sinh viên trả lời 10 câu hỏi đúng – sai?

b) Nếu câu hỏi thuộc loại trả lời sai bị trừ điểm, mỗi câu sinh viên có thể lựa chọn không trả lời. Có bao nhiêu phương án trả lời cho 10 câu hỏi đó?

1.4. Trong Ví dụ 1.6, vẽ sơ đồ đếm các số tự nhiên $\overline{abc}$ chẵn có các chữ số khác nhau, theo các cách ứng với thứ tự chọn

1) a, b, c 2) a, c, b 3) b, a, c 4) b, c, a 5) c, b, a

Cho biết cách nào đơn giản nhất và kinh nghiệm đạt được.

1.5. Xác định giá trị của biến `counter` sau khi thực hiện đoạn chương trình? Cho biết bạn đang vận dụng quy tắc đếm nào?

```
counter := 0
for i := 1 to 12 do
    counter := counter + 1
for j := 5 to 10 do
    counter := counter + 2
for k := 15 downto 8 do
    counter := counter + 3
```

1.6. Trong đoạn chương trình

```
for i := 1 to 12 do
    for j := 5 to 10 do
        for k := 15 downto 8 do
            print (i-j) * k
```

lệnh `print` thực hiện bao nhiêu lần? Áp dụng quy tắc đếm nào?

1.7. Lập trình tìm các số có ba chữ số $\overline{abc}$ sao cho $\overline{abc} = a! + b! + c!$.

1.8. Trong Ví dụ 1.6, lập trình liệt kê các số tự nhiên chẵn có ba chữ số khác nhau.

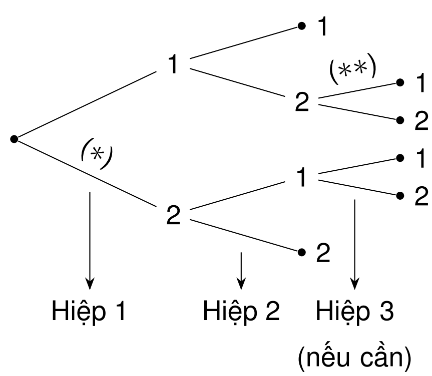
1.2 Biểu đồ cây

Biểu đồ cây là một trường hợp của cấu trúc *cây* tổng quát. Cây và đồ thị là các cấu trúc quan trọng trong khoa học máy tính và lý thuyết tối ưu.

Cây bao gồm một gốc, các nhánh xuất phát từ gốc và nhánh con xuất phát từ điểm cuối của nhánh khác. Biểu diễn mỗi lựa chọn bằng một nhánh, ghi kết quả tại điểm cuối của nhánh đó. Tổng số nhánh không có nhánh con là số phần tử đếm được thỏa mãn kết quả.

Ví dụ 1.11. Vẽ biểu đồ cây biểu diễn các trường hợp có thể của trận đấu quần vợt có 3 hiệp, biết rằng trận đấu kết thúc khi có tay vợt vừa thắng 2 hiệp, và đó là người thắng trận.

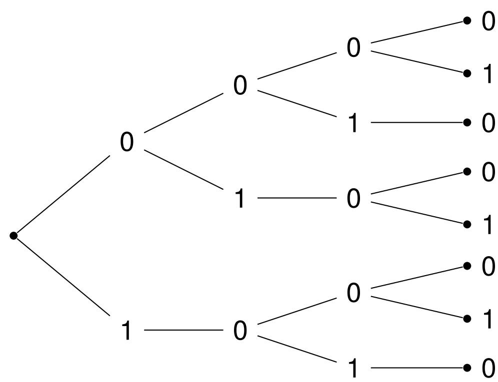
Giải. Đánh số hai tay vợt là 1 và 2. Mỗi hiệp ứng với một tầng của cây, điểm bên phải của mỗi nhánh tại tầng đó đánh dấu người thắng trong hiệp này. Chẳng hạn, nhánh $*$ cho biết người 2 thắng hiệp 1, nhánh $(**)$ ngụ ý người 1 thắng trận đấu vì thắng hiệp 1 và hiệp 3.



□

Ví dụ 1.12. Liệt kê rồi đếm các xâu nhị phân độ dài 4 không có hai số 1 liên tiếp.

Giải. Có 8 xâu như vậy: 0000, 0001, 0010, 0100, 0101, 1000, 1001, 1010.



□

Bài tập 1.2

1.9. Vẽ biểu đồ cây biểu diễn các trường hợp có thể của trận đấu quần vợt có 5 hiệp, biết rằng trận đấu kết thúc khi có tay vợt vừa thắng 3 hiệp, và đó là người thắng trận.

1.3 Hoán vị, chỉnh hợp

Quy tắc nhân được vận dụng để đếm cách sắp xếp tuyến tính của các vật lấy ra từ hệ vật. Khi các vật sắp xếp là phân biệt, ta có các khái niệm *hoán vị* và *chỉnh hợp*.

1.3.1 Hoán vị

Định nghĩa 1.1. *Cho n vật phân biệt. Mỗi cách sắp xếp các vật này gọi là một hoán vị của n vật.*

Một hoán vị của $a_1, a_2, \dots, a_n$ là một bộ có thứ tự $(a_{i_1}, a_{i_2}, \dots, a_{i_n})$. Nếu không gây nhầm lẫn, có thể viết $a_{i_1} a_{i_2} \dots a_{i_n}$.

Định nghĩa 1.2. *Với số tự nhiên n, n giai thừa, ký hiệu n!, xác định bởi*

$$\begin{aligned} 0! &= 1, \\ n! &= 1 \cdot 2 \cdot 3 \cdots (n-2)(n-1)n, \ \text{với } n \geq 1. \end{aligned}$$

Số hoán vị của n vật phân biệt là

$$\underset{\text{vị trí 1}}{n} \times \underset{\text{vị trí 2}}{(n-1)} \times \underset{\text{vị trí 3}}{(n-2)} \times \cdots \times \underset{\text{vị trí n}}{1} = n!. \tag{1.1}$$

Ví dụ 1.13. Có bao nhiêu từ sinh ra từ hoán vị của từ COMPUTER?

Giải. Từ COMPUTER gồm 8 chữ cái phân biệt, nên số hoán vị của nó là

$$8! = 40\,320$$

□

Để tính $n!$, có thể lập trình hoặc dùng gói lệnh/thư viện

Cách 1: Phương pháp lặp

```python
def factorial(n):
    p = 1
    for i in range(1, n+1):
        p *= i
    return p

factorial(8)   # → 8! = 40 320
```

Cách 2: Lập trình, phương pháp đệ quy[*]

```python
def factorial(n):
    if n == 0:
        return 1
    return n * factorial(n-1)

factorial(8)   # → 8! = 40 320
```

Phương pháp đệ quy là lĩnh vực cơ bản của toán rời rạc và phân tích thuật toán. Phương pháp này nảy sinh khi ta muốn giải bài toán bằng cách chia nhỏ, hoặc đưa nó về các bài toán tương tự nhưng cỡ nhỏ hơn. Trong nhiều ngôn ngữ lập trình, điều này được thực hiện bằng cách sử dụng hàm và thủ

[*]có thể khai báo hàm bằng từ khóa `lambda`

```python
factorial = lambda n: 1 if n == 0 else n * factorial(n-1)
```

tục đệ quy, theo quy cách được phép gọi chính nó. Phương pháp này được đề cập nhiều trong chương này và các chương 4, 9.

Cách 3: Thư viện `sympy`, lệnh/hàm `factorial`. Thường sau này ta viết `sympy.factorial`

```
from sympy import *  # nạp trực tiếp tất cả các lệnh của thư viện sympy
factorial(8)  # → 8! = 40320
```

Ví dụ 1.14. a) Liệt kê các hoán vị của các số 1, 2, 3; của 1, 2, 3, 4.

b) Nêu một quy tắc liệt kê các hoán vị của n số.

c) Từ ý tưởng ở ý (b), viết chương trình liệt kê các hoán vị của n số.

Giải. a) Các hoán vị của

1, 2, 3: **1**23 **1**32
213 **2**31
312 **3**21

trong đó hàng đầu gồm các hoán vị bắt đầu là 1, theo sau là các hoán vị của 2 và 3.

1, 2, 3, 4: **1**234 1243 1324 1342 1423 1432
2134 2143 2314 2341 2413 2431
3124 3142 3214 3241 3412 3421
4123 4132 4213 4231 4312 4321

trong đó hàng đầu gồm các hoán vị bắt đầu là 1, theo sau là các hoán vị của 2, 3, và 4.

b) Với các sắp xếp các hoán vị như ý (a), ta thấy để liệt kê các hoán vị của n số, ta thực hiện hai bước

Bước 1: Chọn vị trí đầu tiên cho hoán vị, là một trong n số.

Bước 2: Ứng với mỗi số ở vị trí đầu tiên chọn được, viết vào sau nó hoán vị của $n-1$ số còn lại.

c) Đầu vào của chương trình là một dãy. Với mô tả ở ý (a), ta đặt cấu hình của đầu ra là dãy các hoán vị, trong đó mỗi hoán vị lại là một dãy.

Đầu vào : `a = [1, 2, 3]`
Đầu ra : `[[1, 2, 3], [1, 3, 2], [2, 1, 3],..., [3, 2, 1]]`
Đầu vào : `a = [5]`
Đầu ra : `[[5]]` tức là `[a]`

```
def permutations(a):
    n = len(a)
    if n == 1:
        return [a]
    P = []
    for i in range(n):
        b = a.copy()
        x = b.pop(i)
        for p in permutations(b):
            p = [x] + p
            P.append(p)
    return P
```

□

Một cách khác để liệt kê các hoán vị của 1, 2, 3, ta dùng lệnh `permutations` của thư viện `itertools`

```
import itertools

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

1.3.2 Chỉnh hợp

Định nghĩa 1.3. *Cho n vật phân biệt, và số nguyên r, $1 \leq r \leq n$. Mỗi cách sắp xếp*

a) r vật, có lặp, lấy ra từ n vật gọi là một chỉnh hợp lặp chập r của n vật.

b) r vật phân biệt lấy ra từ n vật gọi là một chỉnh hợp chập r từ n vật.

Đặc biệt, với $r = n$, mỗi chỉnh hợp chập n của chính n vật đó là một hoán vị của n vật này.

Một chỉnh hợp (lặp / phân biệt) của $a_1, a_2, \dots, a_n$ là một bộ có thứ tự $(a_{i_1}, a_{i_2}, \dots, a_{i_r})$. Nếu không gây nhầm lẫn, có thể viết $a_{i_1} a_{i_2} \dots a_{i_r}$.

Mỗi xâu nhị phân độ dài n là một chỉnh hợp lặp chập n của hai số 0 và 1.

Số chỉnh hợp lặp chập r của n vật là

$$\underset{\textbf{vị trí 1}}{n} \times \underset{\textbf{vị trí 2}}{n} \times \underset{\textbf{vị trí 3}}{n} \times \cdots \times \underset{\textbf{vị trí r}}{n} = n^r. \tag{1.2}$$

Số chỉnh hợp* chập r của n vật là

$$\begin{aligned} P(n,r) &= \underset{\textbf{vị trí 1}}{n} \times \underset{\textbf{vị trí 2}}{(n-1)} \times \underset{\textbf{vị trí 3}}{(n-2)} \times \cdots \times \underset{\textbf{vị trí r}}{(n-r+1)} \\ &= n(n-1)(n-2)\cdots(n-r+1) \times \frac{(n-r)(n-r-1)\cdots 2\cdot 1}{(n-r)(n-r-1)\cdots 2\cdot 1} \\ &= \frac{n!}{(n-r)!}. \end{aligned} \tag{1.3}$$

Ví dụ 1.15. a) Với số nguyên dương n, có bao nhiêu xâu nhị phân độ dài n?

b) Liệt kê các xâu nhị phân độ dài $n = 2, 3, 4$.

c) Với $n \geq 2$, cho một quy tắc xây dựng các xâu nhị phân độ dài n từ các xâu nhị phân độ dài $n - 1$.

d) Từ ý tưởng ở ý (c), viết chương trình liệt kê các xâu nhị phân độ dài n.

Giải. a) Mỗi xâu nhị phân độ dài n là một chỉnh hợp lặp chập n của hai số 0 và 1, nên số xâu nhị phân độ dài n là 2^n.

b) Các xâu nhị phân có độ dài

2: **0**0 **0**1

10 **1**1

trong đó hàng đầu gồm các xâu bắt đầu là 0, phần còn lại là xâu độ dài 1.

---

* $P(n, r)$ còn có ký hiệu khác là A_n^r. P là viết tắt của “permutation”, A viết tắt của “arrangement”

3: **0**00 001 010 011
100 101 110 111
trong đó hàng đầu gồm các xâu bắt đầu là 0, phần còn lại là xâu độ dài 2.

4: **0**000 0001 0010 0011 0100 0101 0110 0111
1000 1001 1010 1011 1100 1101 1110 1111
trong đó hàng đầu gồm các xâu bắt đầu là 0, phần còn lại là xâu độ dài 3.

c) Từ mô tả ở ý (b), với $n \geq 2$, để xây dựng các xâu nhị phân độ dài n, ta xếp 0 hoặc 1 vào trước các xâu nhị phân độ dài $n - 1$.

d) Đầu vào : $n = 2$
Đầu ra : `['00', '01', '10', '11']`
Đầu vào : $n = 1$
Đầu ra : `['0', '1']`

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

binary_strs(3)  # ['000', '001', '010', '011', '100', '101', '110', '111']
```

□

Trong Python, để liệt kê xâu nhị phân độ dài 3, có thể dùng thư viện `itertools`

```python
import itertools
list(itertools.product([0, 1], repeat=3))  # [(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)]
```

Ví dụ 1.16. a) Có bao nhiêu từ gồm 5 chữ cái trích ra trong từ COMPUTER?

b) Có bao nhiêu cách chọn 4 người trong 10 người và xếp thành hàng để chụp ảnh?

Giải. a) Mỗi từ 5 chữ cái trích ra trong từ COMPUTER là một chỉnh hợp chập 5 của 8 chữ cái. Số từ đó là

$$P(8,5) = \frac{8!}{(8-5)!} = 6\,720.$$

Cách 1: Lập trình

```python
def P(n, r):
    p = 1
    for i in range(r):
        p *= n - i
    return p

P(8, 5)   # → 6720
```

Cách 2: Thư viện sympy

```python
from sympy import *

n, r = 8, 5
factorial(n) / factorial(n-r)
```

b) Mỗi cách xếp 4 người trong 10 người là một chỉnh hợp chập 4 của 10 người. Số cách là

$$P(10,4) = \frac{10!}{(10-4)!} = 5\,040.$$

□

Ví dụ 1.17. a) Liệt kê các chỉnh hợp chập 3 của 1, 2, 3, 4.

b) Cho một quy tắc liệt kê chỉnh hợp chập r từ chỉnh hợp chập $r-1$, với $r \geq 2$.

c) Từ ý tưởng ở ý (b), viết chương trình liệt kê các chỉnh hợp chập r của n số.

Giải. a) Các chỉnh hợp chập 3 của 1, 2, 3, và 4

123	124	132	134	142	143
213	214	231	234	241	243
312	314	321	324	341	342
412	413	421	423	431	432

trong đó hàng đầu gồm các chỉnh hợp bắt đầu bởi 1, sau nó là chỉnh hợp chập 2 của 2, 3, và 4.

b) Từ mô tả ở ý (a), để xây dựng các chỉnh hợp chập r của n số, với $r \geq 2$:

Bước 1: Chọn vị trí đầu cho chỉnh hợp chập r, là một trong n số đó.

Bước 2: Với mỗi vị trí đầu đã chọn, phần phía sau là chỉnh hợp chập $r - 1$ của $n - 1$ số còn lại.

c) Đầu vào : `a = [1, 2, 3, 4]`, $r = 3$
Đầu ra : `[[1, 2, 3], [1, 2, 4], [1, 3, 2],..., [4, 3, 2]]`
Đầu vào : `a = [2, 4, 7]`, $r = 1$
Đầu ra : `[[2], [4], [7]]`

```python
def permutations(a, r):
    if r == 1:
        return [[i] for i in a]
    P = []
    n = len(a)
    for i in range(n):
        b = a.copy()
        x = b.pop(i)
        for p in permutations(b, r-1):
            p = [x] + p
            P.append(p)
    return P

permutations([1, 2, 3, 4], 3)
```

□

Một cách khác để liệt kê các chỉnh hợp chập 3 của 1, 2, 3, và 4, ta dùng thư viện `itertools`

```python
import itertools
list( itertools.permutations([1, 2, 3, 4], 3) )
```

Bài tập 1.3

1.10. Các chữ cái a, c, f, g, i, t w, x có bao nhiêu

a) hoán vị?

b) hoán vị có chữ đầu là t?

c) hoán vị có chữ đầu là t, và chữ cuối là c?

1.11. Có bao nhiêu từ có ba chữ cái khác nhau trong các chữ cái m, r, a, f và t?

1.12. Có bao nhiêu cách xếp các ký hiệu a, b, c, d, e, e, e, e, e sao cho không có hai ký hiệu e cạnh nhau?

1.13. Tính

a) $P(7,2)$

b) $P(8,4)$

c) $P(10,7)$

d) $P(12,3)$

1.14. Cho các số nguyên $n > r \geq 0$. Chứng minh $P(n,r) = \frac{n}{n-r}P(n-1,r)$.

1.15. Tìm n thỏa mãn

a) $P(n,2) = 90$

b) $P(n,3) = 3P(n,2)$

c) $2P(n,2) + 50 = P(2n,2)$

1.16. Liệt kê các hoán vị của

a) FIT

b) HUCE

1.17. Cho các số nguyên dương $n \geq r$. Lập trình liệt kê các chỉnh hợp lặp chập r của $1, 2, \ldots, n$. [*Gợi ý:* đệ quy, Ví dụ 1.15]

1.18. * Lập trình liệt kê tất cả biển số xe trong hai tình huống ở Ví dụ 1.4.

1.4 Tổ hợp

Định nghĩa 1.4. *Cho n vật phân biệt. Mỗi cách chọn r vật từ n vật đó, không quan tâm thứ tự, gọi là một tổ hợp chập r của n vật.*
Số tổ hợp chập r của n vật ký hiệu là $\binom{n}{r}$*, *và đọc là "n chập r".*

Mỗi tổ hợp của $a_1, a_2, \ldots, a_n$ có dạng $\{a_{i_1}, a_{i_2}, \ldots, a_{i_r}\}$ trong đó không phân biệt thứ tự các vật. Nếu $a_{i_1} < a_{i_2} < \ldots < a_{i_r}$, ta có thể viết gọn tổ hợp đó là $a_{i_1} a_{i_2} \ldots a_{i_r}$ mà không gây nhầm lẫn.

Ta đếm tổ hợp thông qua chỉnh hợp. Để lập chỉnh hợp chập r của n vật, ta thực hiện hai bước

1) Chọn r vật từ n vật. Có $\binom{n}{r}$ cách.

2) Ứng với r vật đã chọn ra ở bước (1), sắp xếp, tức là hoán vị, chúng để thu được một chỉnh hợp chập r của n vật. Số hoán vị của r vật này là $n!$.

Theo quy tắc nhân, $P(n, r) = \binom{n}{r} \times r!$. Suy ra

$$\binom{n}{r} = \frac{P(n,r)}{r!} = \frac{n(n-1)(n-2)\cdots(n-r+1)}{r!} = \frac{n!}{r!\,(n-r)!}. \tag{1.4}$$

Ta thấy, với $n \geq 0$, $C(n, 0) = C(n, n) = 1$. Thêm nữa, với $n \geq 1$, $C(n, 1) = C(n, n-1) = n$. Đặc biệt, khi $0 \leq n < r$, thì $\binom{n}{r} = 0$. Ta cũng có $\binom{n}{r} = \binom{n}{n-r}$ với $n \geq r \geq 0$.

Ví dụ 1.18. Tính $\binom{10}{4}$.

Giải. $\binom{10}{4} = \dfrac{10!}{4!\,(10-4)!} = 210.$

Cách 1: Lệnh `sympy.binomial`

```
from sympy import *

binomial(10, 4)
```

*Ký hiệu khác: $C(n, r)$ hoặc C_n^r. C viết tắt của "combination"

Cách 2: Lập trình, phương pháp lặp

```
1 def binomial(n, r):
2     p = 1
3     for i in range(r):
4         p = p * (n-i) // (i+1)
5     return p

6 binomial(10, 4)
```

Chú ý: ở dòng 4, ta viết phép chia thương `//` thay vì phép chia số thực `/`, vì đó là phép chia hết và ta cũng muốn đảm bảo biến p có kiểu số nguyên. Đặc biệt tình huống này không được dùng phép gán `p *= (n-i) // (i+1)`.

□

Ví dụ 1.19. a) Liệt kê các tổ hợp chập 3 của 5 số 1, 2, 3, 4, 5.

b) Với $n > r \geq 1$, nêu một quy tắc liệt kê các tổ hợp chập r của n số.

c) Viết chương trình liệt kê các tổ hợp chập r của n số, với $n \geq r \geq 1$.

Giải. a) Các tổ hợp chập 3 của 5 số 1, 2, 3, 4, 5

123	124	125	134	135	145
234	235	245	345		

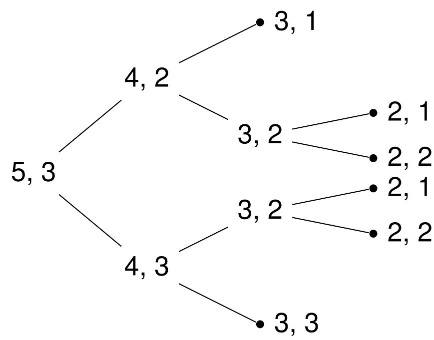
trong đó mỗi tổ hợp ở hàng trên chứa số 1, ghép với tổ hợp chập 2 của 4 số 2, 3, 4, 5, còn hàng dưới là các tổ hợp chập 3 của 4 số này.

b) Từ mô tả ở ý (a), ta xét một số cố định x trong n số đó, thì một tổ hợp chập r của n số có thể là

Trường hợp 1: tổ hợp do ghép x với tổ hợp chập $r - 1$ của $n - 1$ số kia.

Trường hợp 2: tổ hợp chập r của $n - 1$ số kia.

c) Theo ý (a), các tổ hợp 5 chập 3 được xây dựng từ các tổ hợp 4 chập 2 và 4 chập 3, các tổ hợp 4 chập 2 lại cấu thành từ các tổ hợp 3 chập 1 và 3 chập 3,... Ta có cây mô tả quá trình đệ quy để xây dựng các tổ hợp đó



trong đó các nút lá ở các nhánh ngoài cùng bên phải không thể đệ quy được nữa, và ta cần khởi tạo cho chúng. Đó là các trường hợp $r = 1$ hoặc $r = n$.

Input : `a = [1, 2, 3, 4, 5]`, $r = 3$
Output : `[[1, 2, 3], [1, 2, 4], [1, 2, 3],..., [3, 4, 5]]`
Input : `a = [2, 4, 7]`, $r = 1$
Output : `[[2], [4], [7]]` hay `[[i] for i in a]`
Input : `a = [1, 5]`, $r = 2$
Output : `[[1, 5]]` hay `[a]`

```
def combinations(a, r):
    if r == 1:
        return [[i] for i in a]
    n = len(a)
    if r == n:
        return [a]
    C = []
    for c in combinations(a[1:], r-1):
        c = [a[0]] + c
        C.append(c)
    for c in combinations(a[1:], r):
        C.append(c)
    return C

combinations([1, 2, 3, 4, 5], 3)
```

□

Thư viện `itertools` cũng hỗ trợ liệt kê tổ hợp

```python
import itertools
list( itertools.combinations([1, 2, 3, 4, 5], 3) )
```

Ví dụ 1.20. Trong thuật toán sắp xếp nổi bọt ở Ví dụ 1.10, trình bày phương pháp nữa để đếm số phép so sánh khi xếp dãy cỡ n.

Giải. Mỗi phép so sánh (ở dòng 5) được thực hiện tại một cặp (i, j) thỏa mãn

$$\begin{cases} 0 \leq i < n-1 \\ n-1 \geq j > i \end{cases} \Leftrightarrow 0 \leq i < j \leq n-1$$

Cặp (i, j) này là một tổ hợp chập 2 của n số $0, 1, \ldots, n-1$. Vậy số phép so sánh là

$$\binom{n}{2} = \frac{n(n-1)}{2!} = \frac{n(n-1)}{2}.$$

□

Định lý 1.1 (Định lý nhị thức). [*] *Nếu x và y là các biến và n là số nguyên dương, thì*

$$(x+y)^n = \sum_{r=0}^{n} \binom{n}{r} x^r y^{n-r}$$
$$= \binom{n}{0} x^0 y^n + \binom{n}{1} x^1 y^{n-1} + \binom{n}{2} x^2 y^{n-2} + \cdots + \binom{n}{n-1} x^{n-1} y^1 + \binom{n}{n} x^n y^0. \tag{1.5}$$

Trong định lý nhị thức, $\binom{n}{r}$ thường gọi là *hệ số nhị thức* bậc n. Ta cũng có biểu diễn

$$(x+y)^n = \sum_{r=0}^{n} \binom{n}{n-r} x^r y^{n-r} = \sum_{r=0}^{n} \binom{n}{r} x^{n-r} y^r$$
$$= \binom{n}{0} x^n y^0 + \binom{n}{1} x^{n-1} y^1 + \binom{n}{2} x^{n-2} y^2 + \cdots + \binom{n}{n-1} x^1 y^{n-1} + \binom{n}{n} x^0 y^n.$$

[*] Được tìm ra độc lập năm 1665 bởi Isaac Newton, 1643 – 1727, nhà toán học, vật lý, thiên văn học Anh, và năm 1670 bởi James Gregory, 1638 – 1675, nhà toán học, thiên văn học Scotland

Chứng minh. Ta viết $(x+y)^n$ dưới dạng tích

$$(x+y)(x+y)(x+y)\cdots(x+y)(x+y)$$

Áp dụng tính chất phân phối của phép nhân đối với phép cộng, ta được một tổng mà mỗi số hạng là một tích của n thừa số

$$\underbrace{*\ *\ *\cdots\ *\ *}_{n}$$

trong đó mỗi dấu $*$ lần lượt là số hạng x hoặc y trong các dấu ngoặc. Suy ra mỗi số hạng có dạng $x^r y^{n-r}$.

Tiếp theo, ta rút gọn các số hạng đồng dạng, tức là với mỗi r, $0 \le r \le n$, ta đếm trong khai triển có bao nhiêu số hạng có dạng $x^r y^{n-r}$. Có hai bước

1) Chọn r trong n thừa số $x+y$ để lấy số hạng x để cấu thành nên tích, với $\binom{n}{r}$ cách.

2) Chọn $n-r$ thừa số $x+y$ trong $n-r$ thừa số còn lại để lấy số hạng y, với đúng một cách.

Theo quy tắc nhân, số số hạng $x^r y^{n-r}$ là $\binom{n}{r}$. Ta có điều phải chứng minh. □

Ví dụ 1.21. Khai triển $(2x-y)^3$.

Giải.

$$\begin{aligned}
(2x-y)^3 &= \sum_{r=0}^{3}\binom{3}{r}(2x)^{3-r}(-y)^r\\
&= \binom{3}{0}(2x)^3(-y)^0+\binom{3}{1}(2x)^2(-y)^1+\binom{3}{2}(2x)^1(-y)^2+\binom{3}{3}(2x)^0(-y)^3\\
&= 1\cdot 8x^3 * 1 + 3\cdot 4x^2\cdot(-y) + 3\cdot 2x\cdot y^2 + 1\cdot 1\cdot\left(-y^3\right)\\
&= 8x^3 - 12x^2y + 6xy^2 - y^3.
\end{aligned}$$

```
from sympy import *

x, y = symbols('x y')
( (x + y)**2 ).expand()   # (x+y)^2 = x^2 + 2xy + y^2
```

$\square$

Trong Định lý 1.1, lần lượt cho $x = y = 1$ và $x = 1, y = -1$, ta có khẳng định về tổng các hệ số nhị thức.

Hệ quả 1.1. *Với số tự nhiên n,*

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n} = 2^n, \quad \text{và}$$
$$\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \cdots + (-1)^n \binom{n}{n} = 0. \tag{1.6}$$

Hệ quả 1.1 cho ta

$$\binom{n}{0} + \binom{n}{2} + \binom{n}{4} + \cdots = \binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \cdots = 2^{n-1}.$$

Từ Ví dụ 1.19(b), ta có

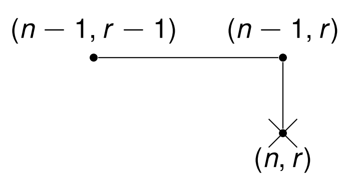
Định lý 1.2 (Hằng đẳng thức Pascal). [*] *Với số nguyên $n > r \geq 1$,*

$$\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}. \tag{1.7}$$

Với $\binom{n}{0} = \binom{n}{n} = 1$, ta có chương trình đệ quy[*]

```python
def binomial(n, r):
    if r == 0 or r == n:
        return 1
    return binomial(n-1, r-1) + binomial(n-1, r)

binomial(10, 4)
```

Lược đồ mô tả hằng đẳng thức Pascal



[*]Blaise Pascal, 1623 – 1662, nhà toán học, vật lý, triết học Pháp

[*]Khai báo bằng từ khóa `lambda`

```python
binomial = lambda n, r: 1 if r==0 or r==n else binomial(n-1, r-1) + binomial(n-1, r)
```

trong đó để tính $\binom{n}{r}$ tại vị trí có dấu $\times$ tại hàng r cột r, ta lấy tổng hai số ở hàng trên, gồm số phía trên và số phía trên bên trái.

Từ đó ta xây dựng tam giác Pascal liệt kê các hệ số nhị thức

$n \backslash r$	0	1	2	3	4	5
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1

Chương trình tính hệ số nhị thức bằng bằng cách cập nhật các hàng của tam giác Pascal, được lập trình theo phương pháp quy hoạch động

```python
def binomial(n, r):
    a = [1]   # hàng 0
    for i in range(1, n+1):   # hàng i
        for j in range(i-1, 0, -1):
            a[j] += a[j-1]
        a.append(1)
    return a[r]

binomial(10, 4)
```

Định lý 1.3 (Hằng đẳng thức Vandermonde)**.** [*] *Cho các số nguyên m, n, r không âm sao cho r không vượt quá m hoặc n. Khi đó*

$$\binom{m+n}{r} = \sum_{k=0}^{r} \binom{m}{k}\binom{n}{r-k}. \tag{1.8}$$

Chứng minh. Vế phải là số cách chọn r vật trong $m+n$ vật.

Mặt khác, chia $m+n$ vật thành hai phần, phần I gồm m vật, và phần II có n vật. Để chọn r vật từ hai phần này, ta chọn k vật từ phần I và $r-k$ vật từ phần II.

[*] Alexandre-Théophile Vandermonde, 1735 – 1796, nhà toán học, nhạc sĩ, nhà hóa học Pháp. Định thức Vandermonde cũng rất nổi tiếng trong đại số tuyến tính.

Với $k = \overline{0, r}$ cố định, theo quy tắc nhân, có $\binom{m}{k} \times \binom{n}{r-k}$ cách chọn r vật. Áp dụng quy tắc cộng, ta có điều phải chứng minh.

□

Bài tập 1.4

1.19. Chứng minh hằng đẳng thức Pascal bằng công thức tổ hợp (1.4).

1.20. Liệt kê và đếm các tổ hợp

a) chập 2 của các chữ a, b, c, d, e và f.

b) chập 3 của các chữ m, r, a, f, và t.

1.21. Tính

a) $C(12, 7)$ b) C_{14}^{12} c) $\binom{15}{10}$

1.22. Cho số nguyên dương $n > 1$. Chứng minh $\binom{n}{2} + \binom{n-1}{2}$ là số chính phương.

1.23. Có bao nhiêu byte có

a) hai số 1?

b) bốn số 1?

c) sáu số 1?

d) ít nhất sáu số 1?

1.24. Có bao nhiêu tam giác tạo nên từ các đỉnh của đa giác lồi n cạnh. Có bao nhiêu tam giác như vậy mà không có cạnh nào là cạnh của đa giác.

1.25. Trong khai triển đầy đủ của $(a + b + c + d)(e + f + g + h)(u + v + w + x + y + z)$ ta được tổng các số hạng như agw, cfx và dgv.

a) Có bao nhiêu số hạng như vậy trong khai triển đầy đủ này?

b) Số hạng nào sau đây không có trong khai triển đầy đủ này?

i) afx ii) bvx iii) chx iv) cgw v) egu vi) dfz

1.26. Tìm hệ số của x^9y^3 trong khai triển của

a) $(x + y)^{12}$ b) $(x + 2y)^{12}$ c) $(2x - 3y)^{12}$

1.27. Cho số nguyên dương n. Tính

a) $\displaystyle\sum_{i=0}^{n} \frac{1}{i!\,(n-i)!}$

b) $\displaystyle\sum_{i=0}^{n} \frac{(-1)^i}{i!\,(n-i)!}$

1.28. Cho hai số nguyên dương m, n. Chứng minh $n\binom{m+n}{m} = (m+1)\binom{m+n}{m+1}$.

1.29. Cho số nguyên dương n. Tính $\binom{n}{0} + 2\binom{n}{1} + 2^2\binom{n}{2} + \cdots + 2^k\binom{n}{k} + \cdots + 2^n\binom{n}{n}$.

1.30. Tìm số thực x thỏa mãn $\displaystyle\sum_{i=0}^{50}\binom{50}{i}8^i = x^{100}$.

1.31. Với số nguyên dương, chứng minh $\displaystyle\binom{2n}{n} = \sum_{k=0}^{n}\binom{n}{k}^2$.

1.5 Hoán vị lặp

Định nghĩa 1.5. *Cho n vật với n_1 vật giống nhau loại 1, n_2 vật giống nhau loại 2,..., và n_r vật giống nhau loại r, trong đó $n_1 + n_2 + \cdots + n_r = n$. Mỗi cách sắp xếp n vật này gọi là một hoán vị lặp của n vật đó.*

Số hoán vị lặp của n vật trên ký hiệu là $\binom{n}{n_1, n_2, \ldots, n_r}$, đọc là "$n$ chập $n_1, n_2, \ldots, n_r$".

Hoán vị là trường hợp đặc biệt của hoán vị lặp khi mỗi loại vật chỉ có 1 vật.

Số hoán vị lặp của n vật trên là

$$\binom{n}{n_1, n_2, \ldots, n_r} = \frac{n!}{n_1!\, n_2! \cdots n_r!}. \tag{1.9}$$

Chứng minh. Để sắp xếp n vật, ta thực hiện các bước sau

1) Chọn n_1 trong n vị trí để xếp vật loại 1, có $\binom{n}{n_1}$ cách.

2) Chọn n_2 trong $n - n_1$ vị trí còn lại để xếp vật loại 2, có $\binom{n-n_1}{n_2}$ cách.

3) Chọn n_3 trong $n - n_1 - n_2$ vị trí còn lại để xếp vật loại 3, có $\binom{n-n_1-n_2}{n_3}$ cách.

...

$n-1$) Chọn n_{r-1} trong $n - n_1 - n_2 - \cdots - n_{r-2}$ vị trí còn lại để xếp vật loại $r-1$, có $\binom{n-n_1-n_2-\cdots n_{r-2}}{n_{r-1}}$ cách.

n) Chọn n_r vị trí trong $n - n_1 - n_2 - \cdots - n_{r-1} = n_r$ vị trí còn lại để xếp vật loại r, có đúng 1 cách.

Theo quy tắc nhân, số hoán vị lặp cần tìm

$$\binom{n}{n_1}\binom{n_2}{n-n_1}\binom{n_3}{n-n_1-n_2}\cdots\binom{n_{r-1}}{n-n_1-\cdots-n_{r-2}}$$
$$= \frac{n!}{n_1!\,(n-n_1)!} \times \frac{(n-n_1)!}{n_2!\,(n-n_1-n_2)!} \times \frac{(n-n_1-n_2)!}{n_3!\,(n-n_1-n_2-n_3)!} \times \cdots \times \frac{(n-n_1-\cdots-n_{r-2})!}{n_{r-1}!\,(n-n_1-\cdots-n_{r-1})!}$$
$$= \frac{n!}{n_1!\,n_2!\,n_3!\cdots n_r!}.$$

□

Tương tự chứng minh trên, ta cũng chỉ ra được

Số cách xếp n vật vào r hộp sao cho hộp i có n_i vật, $1 \le i \le r$, là

$$\binom{n}{n_1, n_2, \ldots, n_r}.$$

Định lý 1.4 (Định lý đa thức)**.** *Với số nguyên dương n, r, hệ số của $x_1^{n_1} x_2^{n_2} \cdots x_r^{n_r}$ trong khai triển của $(x_1 + x_2 + \cdots + x_r)^n$ là*

$$\binom{n}{n_1, n_2, \ldots, n_r},$$

trong đó các số nguyên $n_1, n_2, \ldots, n_r$ không âm và $n_1 + n_2 + \cdots + n_r = n$.

Trong định lý đa thức, các hệ số $\binom{n}{n_1, n_2, \ldots, n_r}$ gọi là hệ số đa thức.

Ví dụ 1.22. Từ MASSASAUGA có bao nhiêu:

a) hoán vị?

b) hoán vị có tất cả chữ A cạnh nhau?

c) hoán vị không có các chữ A cạnh nhau?

Giải. a) Số lượng mỗi chữ cái của từ MASSASAUGA là A: 4, G: 1, M: 1, S: 3, U: 1. Vậy từ có $\dfrac{10!}{4!\,1!\,1!\,3!\,1!} = 25\,200$ hoán vị.

b) Hoán vị có bốn chữ A cạnh nhau là hoán vị của các ký hiệu AAAA, G, M, S, S, S, U. Số hoán vị này là $\dfrac{7!}{1!\,1!\,1!\,3!\,1!} = 840$.

c) Để có hoán vị của từ MASSASAUGA không có chữ A cạnh nhau, ta thực hiện hai bước

1) Lập hoán vị của 6 chữ cái còn lại, gồm $\dfrac{6!}{1!\,1!\,3!\,1!} = 120$ hoán vị. Mỗi hoán vị có dạng $* * * * * *$.

2) Với mỗi hoán vị trên, chèn bốn chữ A vào bốn vị trí đầu hoặc giữa, mỗi vị trí một chữ A, thì sẽ không có chữ A cạnh nhau. Số cách chèn bốn chữ A như vậy là số cách chọn 4 vị trí trong 7 vị trí có thể chèn, và bằng $\dbinom{7}{4} = 35$.

$$\uparrow * \uparrow * \uparrow * \uparrow * \uparrow * \uparrow * \uparrow * \uparrow$$

Theo quy tắc nhân, có $120 \times 35 = 4\,200$ hoán vị của MASSASAUGA không có các chữ A cạnh nhau.

□

Ví dụ 1.23. a) Liệt kê các hoán vị lặp của từ BALL.

b) Cho dãy r vật $a = \left(a_0, a_1, \ldots, a_{r-1}\right)$ gồm các phần tử, và dãy r số tự nhiên $n = \left(n_0, n_1, \ldots, n_{r-1}\right)$. Từ mô tả ở ý (a), nêu ý tưởng liệt kê các hoán vị lặp của các vật trong đó a_i xuất hiện n_i lần.

c) Viết chương trình thể hiện ý tưởng ở ý (b).

Giải. a) Các hoán vị (lặp) của từ BALL

ABLL ALBL ALLB
BALL BLAL BLLA
LABL LALB LBAL LBLA LLAB LLBA

trong đó hàng đầu gồm các hoán vị bắt đầu bằng A, theo sau là hoán vị tạo bởi các chữ còn lại, gồm 1 chữ B và 2 chữ L.

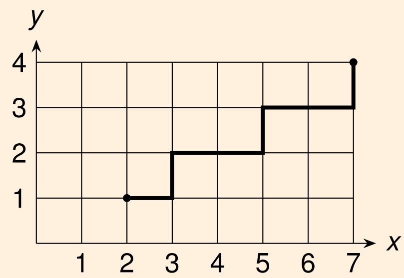
b) Giả sử hoán vị bắt đầu bởi a_i, khi đó phần còn lại là hoán vị của các vật của a với số lần xuất hiện tương ứng là $n\_ = \left(n_0, \ldots, n_{i-1}, n_i - 1, n_{i+1}, \ldots, n_{r-1}\right)$.

c) Input : `a = ['A', 'B', 'L'], n = [1, 1, 2]`
Output : `[['A', 'B', 'L', 'L'], ['A', 'L', 'B', 'L'],...]`
Input : `a = ['A', 'B', 'L'], n = [0, 0, 0]`
Output : `[[]]`

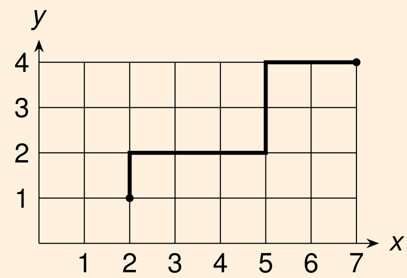
```
def permutations_with_replacement(a, n):
    r = len(a)
    if sum(n) == 0:
        return [[]]
    P = []
    for i in range(r):
        if n[i] > 0:
            n_ = n.copy()
            n_[i] -= 1
            for p in permutations_with_replacement(a, n_):
                p = [a[i]] + p
                P.append(p)
    return P

permutations_with_replacement(['A', 'B', 'L'], [1, 1, 2])
```

□

Ví dụ 1.24. Trên mặt phẳng lưới đơn vị xét các đường đi kiểu bậc thang từ (2, 1) tới (7, 4) sao cho mỗi bước chỉ một đơn vị sang phải (R) hoặc lên trên (U).



(a) R, U, R, R, U, R, R, U



(b) R, U, R, R, U, R, R, U

a) Có bao nhiêu đường đi như vậy?

b) Trình bày ý tưởng liệt kê các đường đi từ điểm (a, b) đến điểm (x, y), với $a \leq x, b \leq y$.

c) Viết chương trình cho ý tưởng ở ý (b).

Giải. a) Trên mỗi đường đi, ta phải sang phải $7 - 2 = 5$ lần và lên trên $4 - 1 = 3$ lần. Như vậy, mỗi đường đi tương ứng với một cách sắp xếp 5 chữ R và 3 chữ U. Số đường đi là

$$\binom{8}{5, 3} = \frac{8!}{5!\,3!} = 56.$$

b) Xuất phát từ (a, b) có hai cách

Cách 1: Đi sang phải tới $(a + 1, b)$, rồi đi tiếp tới (x, y).

Cách 2: Đi lên trên tới $(a, b + 1)$, rồi đi tiếp tới (x, y).

Lưu ý, khi $a = x$, chỉ có 1 cách đi lên trên $y - b$ bước, và khi $b = y$, chỉ có 1 cách đi sang phải $x - a$ bước.

c) Input : $a = 2, b = 1, x = 7, y = 4$
Output : `['RRRRRUUU', 'RRRRURUU', 'RRRRUURU', 'RRRRUUUR', ...]`
Input : $a = 7, b = 1, x = 7, y = 4$
Output : `['UUU']`
Input : $a = 2, b = 4, x = 7, y = 4$
Output : `['RRRRR']`

```
def walks(a, b, x, y):
    if a == x:
        return ['U' * (y-b)]
```

```
    if b == y:
        return ['R' * (x-a)]
    W = []
    for w in walks(a+1, b, x, y):
        w = 'R' + w
        W.append(w)
    for w in walks(a, b+1, x, y):
        w = 'U' + w
        W.append(w)
    return W
```

□

Tổng quát, số cách đi từ điểm (a, b) đến điểm (x, y), với $a \leq x, b \leq y$ là

$$\binom{(x-a)+(y-b)}{x-a, y-b}.$$

Về ý tưởng liệt kê các đường đi, ta cũng có thể xét điểm trước khi tới (x, y) là $(x-1, y)$ hoặc $(x, y-1)$.

Ví dụ 1.25. a) Khai triển $(x+y+z)^3$.

b) Trong khai triển của $(x+y+z)^7$, tìm hệ số của các đơn thức xyz^5, $x^2y^2z^3$, và x^3z^4.

c) Trong khai triển $(a+2b-3c+5)^{10}$, tìm hệ số của a^4bc^3.

Giải. a)

n	**Khai triển**	**Đơn thức**	**Hệ số**
3	$=3+0+0$	$x^3,\ y^3,\ z^3$	$\binom{3}{3,0,0} = \dfrac{3!}{3!\,0!\,0!} = 1$
	$=2+1+0$	$x^2y,\ x^2z,\ xy^2,\ y^2z,\ xz^2,\ yz^2$	$\binom{3}{2,1,0} = \dfrac{3!}{2!\,1!\,0!} = 3$
	$=1+1+1$	xyz	$\binom{3}{1,2,1} = \dfrac{3!}{1!\,1!\,1!} = 6$

Ta được

$$(x+y+z)^3 = x^3+y^3+z^3+3x^2y+3x^2z+3xy^2+3y^2z+3xz^2+3yz^2+6xyz.$$

```python
from sympy import *

x, y, z = symbols('x y z')
( (x + y + z)**7 ).expand()
```

b) Bảng tính hệ số của các đơn thức trong khai triển của $(x + y + z)^7$:

Đơn thức	**Hệ số**
xy^5z	$\binom{7}{1,5,1} = \frac{7!}{1!\,5!\,1!} = 42$
$x^2y^2z^3$	$\binom{7}{2,2,3} = \frac{7!}{2!\,2!\,3!} = 210$
x^3z^4	$\binom{7}{3,0,4} = \frac{7!}{3!\,0!\,4!} = 35$

```python
( (x + y + z)**7 ).expand().coeff(x * y**5 * z)
```

c) Số hạng chứa a^4bc^3 trong khai triển của $(a - 2b + 3c + 5)^{10}$ là

$$\binom{10}{4,1,3,2} a^4(-2b)(3c)^3 5^2.$$

Hệ số cần tìm là $\frac{10!}{4!\,1!\,3!\,2!}(-2)3^3 5^2 = -17\,010\,000.$

```python
a, b, c = symbols('a b c')
expr = (a - 2*b + 3*c + 5)**10
expr.expand().coeff(a**4 * b * c**3)
```

□

Bài tập 1.5

1.32. Liệt kê và đếm các hoán vị của từ COOL.

1.33. Từ DATABASES có bao nhiêu

a) hoán vị?

b) hoán vị có tất cả chữ A cạnh nhau?

c) hoán vị có tất cả chữ A và S xếp liền nhau?

d) không có chữ A cạnh nhau?

1.34. Từ DATAGRAM[*] có bao nhiêu

a) hoán vị?

b) hoán vị có tất cả chữ A cạnh nhau?

1.35. Từ SOCIOLOGICAL có bao nhiêu

a) hoán vị?

b) hoán vị có A và G cạnh nhau?

c) hoán vị có các nguyên âm xếp liền nhau?

1.36. Có bao nhiêu cách xếp các chữ cái trong từ MISSISSIPPI để không có chữ S nào cạnh nhau?

1.37. Có bao nhiêu số nguyên viết từ các chữ số 3, 4, 4, 5, 5, 6, 7 và lớn hơn 5 triệu.

1.38. Có bao nhiêu cách đi trên lưới nguyên Oxy từ

a) $(0, 0)$ đến $(7, 7)$

b) $(2, 7)$ đến $(9, 14)$

1.39. a) Trên lưới nguyên trong không gian $Oxyz$, có bao nhiêu cách đi từ $(-1, 2, 0)$ đến $(1, 3, 7)$, nếu chỉ có các cách đi theo lập phương đơn vị và "hướng về đích" như sau

(H) : (x, y, z) đến $(x + 1, y, z)$
(V) : (x, y, z) đến $(x, y + 1, z)$
(A)[*] : (x, y, z) đến $(x, y, z + 1)$

b) Có bao nhiêu cách đi trên lưới nguyên $Oxyz$ từ $(1, 0, 5)$ đến $(8, 1, 7)$?

c) Tổng quát kết quả ý (a) và (b).

1.40. * Lập trình liệt kê các đường đi trên lưới nguyên $Oxyz$, từ điểm (a, b, c) đến (x, y, z), với $a \leq x, b \leq y, c \leq z$. [*Gợi ý:* vận dụng ý tưởng của Bài Ví dụ 1.24.]

1.41. Tìm hệ số của

a) xyz^2 trong $(x + y + z)^4$

b) xyz^2 trong $(w + x + y + z)^4$

c) xyz^2 trong $(2x - y - z)^4$

d) xyz^{-2} trong $(x - 2y + 3z^{-1})^4$

e) $w^3x^2yz^2$ trong $(2w - x + 3y - 2z)^8$

[*] Dữ liệu được truyền trong các khối bit có cấu trúc qua Internet được gọi là datagram
[*] H, V, A viết tắt cho Horizontal, Vertical và Axial

1.42. Tìm hệ số của $w^2x^2y^2z^2$ trong khai triển của

a) $(w + x + y + z + 1)^{10}$

b) $(2w - x + 3y + z - 2)^{12}$

c) $(v + w - 2x + y + 5z + 3)^{12}$

1.43. Tìm tổng các hệ số trong khai triển

a) $(x + y)^3$

b) $(x + y)^{10}$

c) $(x + y + z)^{10}$

d) $(w + x + y + z)^5$

e) $(2s - 3t + 5u + 6v - 11w + 3x + 2y)^{10}$

1.6 Tổ hợp lặp

Định nghĩa 1.6. *Cho n vật. Mỗi cách chọn r vật, có lặp, từ n vật gọi là một tổ hợp lặp chập r của n vật.*

Số tổ hợp lặp chập r của n vật là

$$\binom{n+r-1}{r} = \frac{(n+r-1)!}{r!\,(n-1)!}. \tag{1.10}$$

Chứng minh. Đặt x_i, $1 \leq i \leq n$ là số vật i chọn được. Khi đó mỗi tổ hợp lặp chập r từ n vật tương ứng $1-1$ với một nghiệm nguyên không âm của phương trình

$$x_1 + x_2 + \cdots + x_n = r. \tag{1.11}$$

Mặt khác trong (1.11), ta thay các dấu $+$ bằng dấu $|$, còn số x_i ta thay bằng x_i dấu $*$ ở giữa các dấu $|$ thứ $i-1$ và thứ i, riêng số x_1 thay bằng x_1 dấu $*$ bên trái dấu $|$ thứ nhất, và số x_n thay bằng x_n dấu $*$ bên phải dấu $|$ thứ $n-1$, chẳng hạn nghiệm $(3, 1, 0, \ldots, 2, 0)$ ứng với dãy

$$* \quad * \quad * \quad | \quad * \quad | \quad | \quad * \quad \cdots \quad | \quad * \quad * \quad |$$

tức mỗi nghiệm nguyên không âm của (1.11) lại tương ứng $1-1$ với một cách sắp xếp r dấu $*$ và $n-1$ dấu $|$, chính là hoán vị lặp của chúng. Vậy số tổ hợp lặp chập r của n vật là

$$\binom{r+(n-1)}{r,\, n-1} = \frac{(r+(n-1))}{r!\ (n-1)!} = \binom{n+r-1}{r}.$$

□

Xét bài toán chia r vật giống nhau vào n hộp khác nhau. Đặt x_i, $1 \leq i \leq n$ là số vật đặt vào hộp i. Khi đó mỗi cách chia tương ứng $1-1$ với một nghiệm nguyên không âm của (1.11), gọi là một *phân $n-$hạng không âm của r*.

Số tổ hợp lặp chập r của n vật cũng là:

a) số cách phân $n-$hạng không âm của r, hay số nghiệm nguyên của phương trình

$$x_1 + x_2 + \cdots + x_n = r, \quad x_i \geq 0, 1 \leq i \leq n.$$

b) số cách xếp r vật giống nhau vào n hộp khác nhau.

Ví dụ 1.26. a) Có bao nhiêu cách chọn 4 sản phẩm từ 3 loại sản phẩm A, B, C?

b) Liệt kê các cách chọn ở ý (a).

c) Viết chương trình liệt kê các tổ hợp lặp chập r của n vật bằng cách liệt kê các cách phân $n-$hạng không âm của r.

Giải. a) Số cách chọn 4 sản phẩm từ 3 loại sản phẩm A, B, C là số tổ hợp lặp chập 4 của 3 vật

$$\binom{3+4-1}{4} = 15$$

b) Các cách chọn 4 sản phẩm từ 3 loại sản phẩm A, B, C

{4A}				
{3A, B}	{3A, C}			
{2A, 2B}	{2A, B, C}	{2A, 2C}		
{A, 3B}	{A, 2B, C}	{A, B, 2C}	{A, 3C}	
{4B}	{3B, C}	{2B, 2C}	{B, 3C}	{4C}

trong đó theo từng hàng, số sản phẩm A được chọn giảm dần, và trên mỗi hàng, số sản phẩm B giảm dần.

```
import itertools
list( itertools.combinations_with_replacement(['A', 'B', 'C'], 4) )
```

c) Biến đổi

$$\begin{cases} x_1 + x_2 + \cdots + x_n = r \\ x_i \geq 0,\ 1 \leq i \leq n \end{cases} \Leftrightarrow \begin{cases} x_1 = i,\ i = \overline{0, r} \\ x_2 + x_3 + \cdots + x_n = r - i, \quad x_i \geq 0,\ 2 \leq i \leq n \end{cases}$$

tức là ta xây dựng cách phân $n-$hạng không âm từ cách phân $(n-1)-$hạng không âm.

Input : $n = 4, r = 3$
Output : `[[0, 0, 4], [0, 1, 3], [0, 2, 2],...]`
Input : $n = 1, r = 5$
Output : `[[5]]` tức là `[[r]]`

```
def solutions(n, r):
    if n == 1:
        return [[r]]
    S = []
    for i in range(r+1):
        for s in solutions(n-1, r-i):
            s = [i] + s
            S.append(s)
    return S

solutions(3, 4)
```

□

Ví dụ 1.27. Trong đoạn chương trình sau, lệnh `print` thực hiện bao nhiêu lần

```
for i = 1 to n do
    for j = 1 to i do
        for k = 1 to j do
            print i, j, k
```

Giải. Mỗi lệnh `print` ứng với một bộ (i, j, k) được in ra, với $1 \leq k \leq j \leq i \leq n$. Đó chính là tổ hợp lặp chập 3 của n số $1, 2, \dots, n$. Do đó số lệnh `print` được thực hiện là

$$\binom{n+3-1}{3} = \binom{n+2}{3} = \frac{(n+2)(n+1)n}{3!} = \frac{n(n+1)(n+2)}{6}.$$

□

Ví dụ 1.28. a) Khai triển đầy đủ của $(x_1 + x_2 + \cdots + x_r)^n$, tức là tổng của các đơn thức không đồng dạng, có bao nhiêu số hạng?

b) Khai triển đầy đủ của $(x + y + z)^7$ có bao nhiêu số hạng?

Giải. a) Mỗi số hạng trong khai triển đầy đủ của $(x_1 + x_2 + \cdots + x_r)^n$ có dạng

$$\binom{n}{n_1, n_2, \dots, n_r} x_1^{n_1} x_2^{n_2} \cdots x_r^{n_r},$$

với $n_1, n_2, \dots, n_r$ là các số nguyên không âm và $n_1 + n_2 + \cdots + n_r = n$. Vì vậy, số số hạng trong khai triển là $\binom{r+n-1}{n}$.

b) Số số hạng trong khai triển đầy đủ của $(x + y + z)^7$ là

$$\binom{3+7-1}{7} = 36.$$

□

Ở trang 40, ta đã đếm các nghiệm nguyên phương trình $x_1 + x_2 + \cdots + x_n = r$ với $x_i \geq 0$, $1 \leq i \leq n$. Bây giờ ta xét phương trình nghiệm nguyên này với điều kiện cận dưới tổng quát

$$x_i \geq a_i, \tag{1.12}$$

trong đó a_i là số nguyên. Đặt $x_i = x_i' + a_i$, ta được phương trình nghiệm nguyên không âm tương đương

$$x_1' + x_2' + \cdots + x_n' = \underbrace{r - a_1 - a_2 - \cdots - a_n}_{r'}, \tag{1.13}$$

Vì vậy, hai phương trình có có nghiệm bằng nhau và bằng $\binom{n+r'-1}{r'}$.

Ví dụ 1.29. Xác định số nghiệm nguyên không âm thỏa mãn

a) $x_1 + x_2 + \cdots + x_6 = 10$. b) $x_1 + x_2 + \cdots + x_6 < 10$.

Giải. a) Phương trình $x_1 + x_2 + \cdots + x_6 = 10$ có $\binom{6+10-1}{10} = 3\,003$ nghiệm nguyên không âm.

b) **Cách 1:** Mỗi nghiệm nguyên không âm của $x_1 + x_2 + \cdots + x_6 < 10$ tương ứng $1-1$ với một nghiệm nguyên phương trình

$$x_1 + x_2 + \cdots + x_6) + x_7 = 10, \ \ x_i \geq 0, 1 \leq i \leq 6, \ \ \text{và } x_7 \geq 7.$$

Ở đây r' trong (1.13) là

$$r' = 10 \underbrace{-0-0-\cdots-0}_{6} -1 = 9$$

nên số nghiệm cần tìm là $\binom{n+r'-1}{r'} = \binom{7+9-1}{9} = 5\,005.$

Cách 2: Theo quy tắc cộng, số nghiệm nguyên không âm thỏa mãn $x_1 + x_2 + \cdots + x_6 < 10$ là tổng số nghiệm nguyên không âm của các phương trình $x_1 + x_2 + \cdots + x_6 = r$, với $0 \leq r \leq 9$, và bằng

$$\sum_{r=0}^{9} \binom{6+r-1}{r} = 5\,005.$$

```python
from sympy import *

r = symbols('r')
Sum(binomial(6+r-1, r), (r, 0, 9)).doit()
```

□

Trong chương này và Chương 3, 8, 9, ta hay làm việc với khái niệm sau

Định nghĩa 1.7. *Cho số nguyên dương n. Mỗi phân hạng của n là một cách tách n thành tổng của một dãy số nguyên dương $x_1, x_2, \ldots, x_k$, với $k = 1, 2, \ldots$.*

$$n = x_1 + x_2 + \cdots + x_k. \tag{1.14}$$

Ví dụ 1.30. a) Tìm các phân hạng của $n = 1, 2, 3, 4, 5$.

b) Có bao nhiêu phân hạng của n?

c) Lập trình liệt kê các phân hạng của n.

Giải. a) Các phân hạng của 1, 2, 3, 4, 5 là

1: 1

2: **2**
1 + 1

3: **3**
2 + 1
1 + 2   1 + 1 + 1

4: **4**
3 + 1
2 + 2   2 + 1 + 1
1 + 3   1 + 2 + 1   1 + 1 + 2   1 + 1 + 1 + 1

trong đó các phân hạng trên mỗi hàng có cùng số hạng đầu.

b) Phân hạng k số hạng, $k = 1, 2, \ldots, n$, của n

$$x_1 + x_2 + \cdots + x_k = n, \quad \text{với } x_i \geq 1$$

có $\binom{k + (n-k) - 1}{n-k} = \binom{n-1}{n-k}$ nghiệm. Theo quy tắc cộng, số phân hạng của n là

$$\sum_{k=1}^{n} \binom{n-1}{n-k} = \binom{n-1}{n-1} + \binom{n-1}{n-2} + \cdots + \binom{n-1}{1} + \binom{n-1}{0}.$$

Theo Hệ quả 1.1, đây là tổng các hệ số nhị thức bậc $n-1$, và bằng 2^{n-1}.

c) Với $k \geq 2$

$$x_1 + x_2 + \cdots + x_k = n \Leftrightarrow \begin{cases} x_1 = i, \ 1 \leq i \leq n-1 \\ x_2 + x_3 + \cdots + x_k = n - i \end{cases}$$

tức là các phân hạng 2 số hạng trở lên của n được tạo thành từ các phân hạng của các số nhỏ hơn n.

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

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

summands(3)
```

□

Bài tập 1.6

1.44. Tìm số nghiệm nguyên của phương trình $x_1 + x_2 + x_3 + x_4 = 32$ thỏa mãn

a) $x_i \geq 0, 1 \leq i \leq 4$

b) $x_i > 0, 1 \leq i \leq 4$

c) $x_1, x_2 \geq 5, x_3, x_4 \geq 7$

d) $x_i \geq 8, 1 \leq i \leq 4$

e) $x_i \geq -2, 1 \leq i \leq 4$

f) $x_1, x_2, x_3 > 0, 0 < x_4 \leq 25$

1.45. Tìm số nghiệm nguyên của $x_1 + x_2 + x_3 + x_4 + x_5 < 40$ thỏa mãn

a) $x_i \geq 0, 1 \leq i \leq 5$

b) $x_i \geq -3, 1 \leq i \leq 5$

1.46. Tìm số nguyên dương n để các phương trình

$$(1) \quad x_1 + x_2 + \cdots + x_{19} = n, \quad \text{và}$$
$$(2) \quad y_1 + y_2 + \cdots + y_{64} = n, \quad \text{và}$$

có cùng số nghiệm.

1.47. a) Hệ phương trình $x_1 + x_2 + \cdots + x_7 = 37$, $x_1 + x_2 + x_3 = 6$ có bao nhiêu nghiệm nguyên không âm?

b) Có bao nhiêu nghiệm ở ý (a) có $x_1, x_2, x_3 > 0$?

1.48. Xét khai triển của $(3v + 2w + x + y + z)^8$.

a) Tìm hệ số của v^2w^4xz.

b) Có bao nhiêu số hạng trong khai triển?

1.49. Lệnh `print` thực hiện bao nhiêu lần trong đoạn chương trình sau

```
for i = 1 to 20 do
    for j = 1 to i do
        for k = 1 to j do
            for m = 1 to k do
                print i, j, k, m
```

1.50. Tìm giá trị của biến `counter` sau khi thực hiện đoạn chương trình

```
counter = 10
for i = 1 to 15 do
    for j = i to 15 do
        for k = j to 15 do
            for m = k to 15 do
                counter = counter + 1
```

1.51. a) Cho các số nguyên dương $m \geq n$. Chứng minh số cách chia m vật giống nhau vào n hộp khác nhau sao cho không có hộp nào trống là $\binom{m-1}{m-n} = \binom{m-1}{n-1}$.

b) Chứng minh số cách chia m vật giống nhau vào n hộp khác nhau sao cho mỗi hộp chứa ít nhất r vật là $\binom{m-1-(r-1)n}{n-1}$. Ở đây $m \geq rn$.

1.52. Lập trình để liệt kê các các nghiệm nguyên của

a) $x_1 + x_2 + x_3 = 10$, $x_i \geq 0$, $1 \leq i \leq 3$

b) $x_1 + x_2 + x_3 + x_4 = 4$, $x_i \geq -2$, $1 \leq i \leq 4$

1.53. Trong Ví dụ 1.26, ta đã liệt kê nghiệm nguyên không âm của phương trình $x_1 + x_2 + \cdots + x_n = r$, với n nguyên dương, r là số tự nhiên. Bây giờ ta đưa ra một phương pháp liệt kê khác. Xét hai trường hợp

1) $x_1 = 0$, khi đó $x_2 + x_3 + \cdots + x_n = r$.

2) $x_1 > 0$, khi đó $(x_1 - 1) + x_2 + \cdots x_n = r - 1$.

Viết chương trình từ ý tưởng trên.

1.7 Sinh các hoán vị và tổ hợp

Bằng cách đánh số n vật bằng các số từ 1 tới n, ta đưa bài toán tìm hoán vị, tổ hợp của n vật về bài toán tìm hoán vị, tổ hợp của các số 1, 2,..., n.

Định nghĩa 1.8 (Thứ tự từ điển). *Cho hai dãy số nguyên $a_1, a_2, \dots, a_m$ và $b_1, b_2, \dots, b_n$. Ta nói dãy $a_1, a_2, \dots, a_m$ đứng trước (hay nhỏ hơn) dãy $b_1, b_2, \dots, b_n$ nếu*

a) $a_1 = b_1, a_2 = b_2, \dots, a_m = b_m$ và $m < n$; hoặc

b) có chỉ số k nào đó thỏa mãn

$$a_1 = b_1, a_2 = b_2, \dots, a_{k-1} = b_{k-1}, \text{ và } a_k < b_k.$$

Ví dụ 1.31. a) Dãy 4, 1, 2 đứng trước dãy 4, 1, 2, 3; dãy $(3, 1, 2, 5) < (3, 1, 4)$.

b) Trong các hoán vị của 1, 2, 3, 4, 5, hoán vị 23415 đứng trước hoán vị 23514.

c) Trong các tổ hợp chập 4 của 1, 2, 3, 4, 5, 6, tổ hợp 1256 đứng trước 1345.

Chương trình kiểm tra thứ tự từ điển của hai dãy $a = \left(a_0, a_1, \dots, a_{m-1}\right)$ và $b = \left(b_0, b_1, \dots, b_{n-1}\right)$.

```python
def compare(a, b):
    m, n = len(a), len(b)
    i = 0
    while i < m and i < n and a[i] == b[i]:
        i += 1
    if i == m == n:
        return('=')
    if i == m < n:
        return('<')
    if i == n < m:
        return('>')
    if i < m and i < n:
```

```python
        if a[i] < b[i]:
            return('<')
        else:
            return('>')

compare([4, 1, 2], [4, 1, 2, 3])  # → '<'

compare([3, 1, 4], [3, 1, 2, 5])  # → '>'
```

Xét các hoán vị của $1, 2, \ldots, n$. Thuật toán sinh các hoán vị dựa trên thủ tục tìm hoán vị kế tiếp của hoán vị cho trước $a_1 a_2 \ldots a_n$.

1) Xuất phát từ a_n, tìm dãy con đầu tiên không còn tăng nữa:

$$a_n < a_{n-1} < \ldots < a_k > a_{k-1}.$$

2) Hoán vị kế tiếp của $a_1 a_2 \ldots a_n$ thu được bằng cách:

   i) Trong dãy đơn điệu $a_k a_{k+1} \ldots a_n$, tìm số nhỏ nhất $a_i > a_{k-1}$, đổi chỗ hai số này. Khi đó vẫn có $a_k > a_{k+1} > \ldots > a_n$.

   ii) Đổi cho dãy $a_k \ldots a_n$ theo thứ tự ngược lại (tăng).

Ví dụ 1.32. a) Tìm hoán vị liền sau 362541.

b) Chương tìm hoán vị liền sau một hoán vị cho trước của $1, 2, \ldots, n$.

Giải. a) Hoán vị đứng sau hoán vị 362541 là 364125.

$$3 \quad 6 \quad 2 < \boxed{5 > 4 > 1}$$

đổi chỗ min, > 2

$$3 \quad 6 \quad 4 \quad \boxed{5 \quad 2 \quad 1}$$ lật dãy

$$\longrightarrow 3 \quad 6 \quad 4 \quad 1 \quad 2 \quad 5$$

b)

```python
def next_permutations(a):
    n = len(a)
    k = n - 1
    while k >= 1 and a[k-1] > a[k]:
        k -= 1

    if k == 0:
        return None

    i = n - 1
    while a[i] < a[k-1]:
        i -= 1
    a[k-1], a[i] = a[i], a[k-1]

    b = a[k:]
    b.reverse()
    return a[:k] + b

next_permutations([3, 6, 2, 5, 4, 1])  # [3, 6, 4, 1, 2, 5]
```

□

Thủ tục tìm tổ hợp chập r liền sau tổ hợp $a_1 a_2 \dots a_r$ của các số $1, 2, \dots, n$, theo thứ tự từ điển:

1) Từ phải qua trái, tìm số a_i đầu tiên sao cho $a_i \neq n - r + i$.

2) Thay $a_i, \dots, a_r$ bằng các số liên tiếp $a_i + 1, a_i + 2, \dots$

Ví dụ 1.33. a) Tìm tổ hợp liền sau tổ hợp 1256 của 1, 2, 3, 4, 5, 6.

b) Xét các tổ hợp chập r của các số $1, 2, \dots, n$, với $n \geq r \geq 1$. Lập trình tìm tổ hợp liền sau một tổ hợp cho trước.

Giải. a) $n = 6, r = 4$. Tổ hợp liền sau 1256 của 1, 2, 3, 4, 5, 6 là 1345.

a_i	1	$\boxed{2}$	5	6
i	1	2	3	4
$n - r + i = i + 2$		$\boxed{4}$	5	6
a_i'	1	3	4	5

b)

```
def next_combinations(n, a):
    r = len(a)
    i = r - 1
    while i >= 0 and a[i] == n - r + (i + 1):
        i -= 1

    if i == -1:
        return None

    return a[:i] + [a[i] + j for j in range(1, r-i+1)]

next_combinations(6, [1, 2, 5, 6])  # [1, 3, 4, 5]
```

□

Mỗi tổ hợp được biểu diễn bởi một xâu nhị phân độ dài *n*, cũng có thứ tự từ điển. Để tìm xâu liền sau của một xâu nào đó:

1) Từ phải qua trái, tìm vị trí đầu tiên tiên bằng 0.

2) Từ vị trí đó sang phải, thay 0 bởi 1 và 1 bởi 0.

Ví dụ 1.34. a) Tìm xâu nhị phân liền sau xâu 1000 1001 11.

b) Xét các xâu nhị phân độ dài *n*. Lập trình tìm xâu nhị phân liền sau xâu nhị phân cho trước.

Giải. a) Xâu liền sau 1000 10**0**1 11 là 1000 10**10 00**.

b)

```
def next_bin_str(a):
    n = len(a)
    i = n - 1
    while i >=0 and a[i] == 1:
        i -= 1

    if i == -1:
```

```
        return None

    for j in range(i, n):
        a[j] = 1 - a[j]
    return a

a = [1, 0, 0, 0, 1, 0, 0, 1, 1, 1]
next_bin_str(a)  # [1, 0, 0, 0, 1, 0, 1, 0, 0, 0]
```

□

Bài tập 1.7

1.54. Xếp các hoán vị của 1, 2, 3, 4, 5 theo thứ tự từ điển: 43521, 15432, 45321, 23451, 23514, 14532, 21345, 45213, 31452, 31542.

1.55. Xếp các hoán vị của 1, 2, 3, 4, 5, 6 theo thứ tự từ điển: 234561, 231456, 165432, 156423, 543216, 541236, 231465, 314562, 432561, 654321, 654312, 435612.

1.56. Tìm hoán vị đứng sau các hoán vị sau theo thứ tự từ điển

a) 1432
b) 54123
c) 12453
d) 45231
e) 6714235
f) 31528764

1.57. Xét các hoán vị của $1, 2, \ldots, n$. Tận dụng chương trình trong Ví dụ 1.32(b), liệt kê các hoán vị đứng sau một hoán vị cho trước.

1.58. Xét các tổ hợp chập r của $1, 2, \ldots, n$, với $n \geq r \geq 1$. Từ chương trình ở Ví dụ 1.33(b), lập trình liệt kê các tổ hợp đứng sau tổ hợp cho trước.

1.8 Số Catalan (đang cập nhật)

Số Catalan[*] xuất hiện trong nhiều lĩnh vực toán học, khoa học máy tính và có nhiều ứng dụng thực tế. Ngoài các ví dụ sau, bạn đọc có thể tìm hiểu Ví dụ 8.18, và nhiều hơn trong [5].

[*] Engène Charles Catalan, 1814 – 1894, nhà toán học Bỉ

Trong Ví dụ 1.24, ta đã đếm số cách đi giữa hai điểm trên lưới đơn vị. Xét trường hợp đặc biệt sau.

Ví dụ 1.35. Cho số nguyên dương n, xét các cách đi trên lưới nguyên từ điểm $(0, 0)$ tới (n, n) sao cho không đi lên phía trên đường $y = x$.

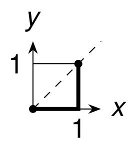
a) Liệt kê các cách đi đó ứng với $n = 1, 2, 3$.

b) Đếm số cách đi đó với n bất kỳ.

c) Nêu ý tưởng và viết chương trình liệt kê các cách đi đó với n bất kỳ.

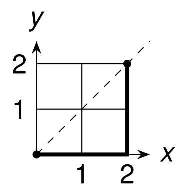
Giải. a)

$n = 1$:

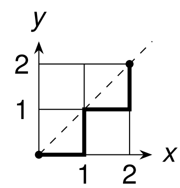


RU

$n = 2$:

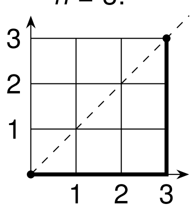


RRUU

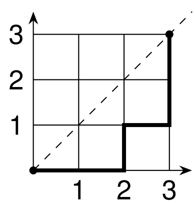


RURU

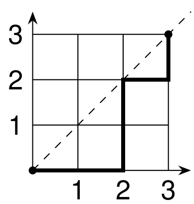
$n = 3$:



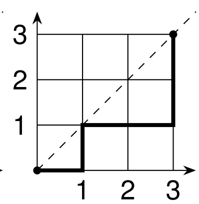
RRRUUU



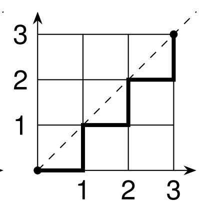
RRURUU



RRUURU



RURRUU



RURURU

b) Mỗi cách đi từ $(0, 0)$ tới (n, n) ứng với một xâu độ dài $2n$ gồm n chữ/bước R và n bước U. Số cách đi là hoán vị lặp của $2n$ bước đó, và bằng

$$\binom{2n}{n, n} = \frac{(2n)!}{n!\ n!} = \binom{2n}{n}.$$

Đặt r_k, u_k, $1 \le k \le 2n$, tương ứng là số bước R, U trong cách đi sau k bước đầu.

Với một cách đi lỗi thì tại thời điểm nào đó, số bước U nhiều hơn số bước R, tức là sau bước k nào đó, $1 \le k < 2n$, $u_k > r_k$. Hơn nữa, nếu k là số bước đầu tiên gây ra lỗi, thì $u_k = r_k + 1$.

1) Xét một cách đi lỗi. Đặt k là số bước đầu tiên gây ra lỗi, tức là $u_k = r_k+1$. Trong các bước sau, số bước R và U tương ứng là $n-r_k$ và $n-u_k$. Ta xây dựng cách đi mới từ cách đi lỗi như sau

   i) k bước đầu giữ nguyên, gồm r_k bước R và u_k bước U.

   ii) Các bước sau thay R bởi U, và U bởi R. Đoạn này gồm $n-u_k$ bước R và $n-r_k$ bước U.

   Suy ra, trong cách đi mới, số bước R

$$r_k + (n-u_k) = n-(u_k-r_k) = n-1,$$

   và số bước U là

$$u_k + (n-r_k) = n+(u_k-r_k) = n+1.$$

   Như vậy, mỗi cách đi lỗi tương ứng với một cách đi gồm $n-1$ bước R và $n+1$ bước U.

2) Ngược lại, xét một cách đi gồm $n-1$ bước R và $n+1$ bước U. Khi đó trong $k = 2n-1$ bước đầu $u_k \geq n > n-1 \geq r_k$.

   Đặt k, $1 \leq k \leq 2n-1$, là số bước nhỏ nhất thỏa mãn $u_k > r_k$. Khi đó $u_k = r_k+1$, vì $u_1 - r_1 = \pm 1$, và sau mỗi bước $u_k - r_k$ chỉ tăng hoặc giảm 1. Xây dựng cách đi mới từ cách đi này, tương tự trên, như sau

   i) k bước đầu giữ nguyên, gồm r_k bước R và u_k bước U.

   ii) Các bước sau thay R bởi U, và U Bởi R. Đoạn này gồm $(n+1)-u_k$ bước R, và $(n-1)-r_k$ bước U.

   Khi đó cách đi mới có số bước R là

$$r_k + (n+1) - u_k = n + (r_k+1) - u_k = n,$$

   số bước U là

$$u_k + (n-1) - r_k = n + u_k - (u_k+1) = n,$$

   và có $u_k = r_k+1 > r_k$ nên đó là cách đi lỗi. Như vậy, mỗi cách đi gồm $n-1$ bước R và $n+1$ bước U tương ứng với một cách đi lỗi.

Từ hai lập luận (1) và (2), mỗi cách đi lỗi tương ứng 1– với một cách đi gồm $n-1$ bước R và $n+1$ bước U, và số cách đi này là

$$\binom{2n}{n-1, n+1} = \frac{(2n)!}{(n-1)!\,(n+1)!} = \binom{2n}{n-1} = \frac{n}{n+1}\frac{(2n)!}{n!\,n!} = \frac{n}{n+1}\binom{2n}{n}.$$

Theo quy tắc cộng, số cách đi đúng là

$$\binom{2n}{n} - \frac{n}{n+1}\binom{2n}{n} = \frac{1}{n+1}\binom{2n}{n}.$$

c) Tương tự Ví dụ 1.24, để đi từ (a, b) đến (n, n), xét hai trường hợp

Trường hợp 1: $a = b$, thì chỉ có cách đi sang phải tới $(a+1, b)$.

Trường hợp 2: $a > b$, thì cũng có hai cách đi, sang phải tới $(a+1, b)$ và lên trên $(a, b+1)$.

```
def catalan_walks(a, b, n):
    if a == n:
        return ['U' * (n-b)]
    W = []
    if a == b:
        for w in catalan_walks(a+1, b, n):
            w = 'R' + w
            W.append(w)
    if a > b:
        for w in catalan_walks(a+1, b, n):
            w = 'R' + w
            W.append(w)
        for w in catalan_walks(a, b+1, n):
            w = 'U' + w
            W.append(w)
    return W

catalan_walks(0, 0, 3)
```

□

Số

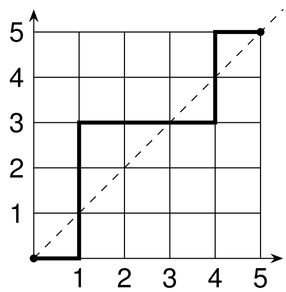
$$C_n = \frac{1}{n+1}\binom{2n}{n} \tag{1.15}$$

gọi là số Catalan. Vài số Catalan đầu tiên

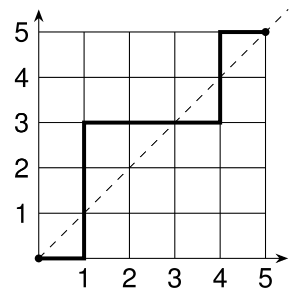
n	0	1	2	3	4	5	6	7	8	9	10
C_n	1	1	2	5	14	42	132	429	1430	4862	16796

```python
from sympy import *
[binomial(2*n, n) / (n+1) for n in range(11)]
```

Minh họa lập luận ở ý (b) trong trường hợp $n = 5$.



Cách đi lỗi: **R**, **U**, **U**, U, R, R, R, U, U, R



Cách đi mới: **R**, **U**, **U**, R, U, U, U, R, R, U

trong đó tương ứng $1-1$ được mô tả như sau

$$\begin{array}{ccc} & \boxed{\text{R} \quad \text{U} \quad \text{U}} & \boxed{\text{U} \quad \text{R} \quad \text{R} \quad \text{R} \quad \text{U} \quad \text{U} \quad \text{R}} \\ & k \text{ min} : u_k > r_k & \text{thay R bởi U, U bởi R} \\ \xleftrightarrow{1-1} & \boxed{\text{R} \quad \text{U} \quad \text{U}} & \boxed{\text{R} \quad \text{U} \quad \text{U} \quad \text{U} \quad \text{R} \quad \text{R} \quad \text{U}} \end{array}$$

Cách tìm số Catalan trong ví dụ trên có thể vận dụng trong một số tình huống sau:

1. Số cách xếp n số 1 và n số -1 thành một dãy sao cho mọi dãy con đầu của dãy đều có tổng không âm, là C_n.

   Thật vậy, để dãy con đầu có tổng không âm, thì số số hạng -1 trong đó phải không vượt quá số số 1.

   Với $n = 3$, có $C_5 = 5$ cách xếp như vậy

   $1, 1, 1, -1, -1, -1$ $\qquad$ $1, 1, -1, -1, 1, -1$ $\qquad$ $1, -1, 1, -1, 1, -1$

   $1, 1, -1, 1, -1, -1$ $\qquad$ $1, -1, 1, 1, -1, -1$

2. Số xâu nhị phân gồm n số 0 và n số 1 sao cho trong đoạn xâu đầu bất kỳ, số số 0 không vượt quá số số 1, là C_n.

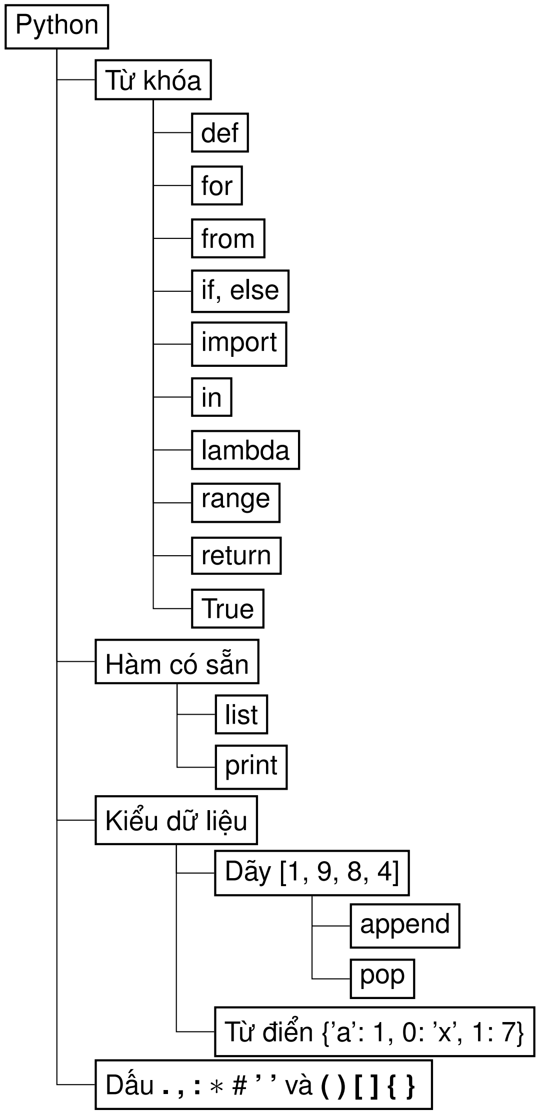
   Với $n = 4$, có $C_4 = 14$ xâu độ dài 8 như vậy

11110000	11100010	11010010	10111000	10101100
11101000	11011000	11001100	10110100	10101010
11100100	11010100	11001010	10110010	

1.9 Tóm tắt

Bảng đếm các cách chọn r vật từ n vật:

Thứ tự	**Lặp**	**Kiểu**	**Công thức**	
có	có	chỉnh hợp lặp	n^r	$n, r \geq 0$
có	không	chỉnh hợp	$P(n,r) = \dfrac{n!}{(n-r)!}$	$0 \leq r \leq n$
không	có	tổ hợp lặp	$\dbinom{n+r-1}{r}$	$n, r \geq 0$
không	không	tổ hợp	$\dbinom{n}{r} = \dfrac{n!}{r!\,(n-r)!}$	$0 \leq r \leq n$


Python
Từ khóa
def
for
from
if, else
import
in
lambda
range
return
True
Hàm có sẵn
list
print
Kiểu dữ liệu
Dãy [1, 9, 8, 4]
append
pop
Từ điển {’a’: 1, 0: ’x’, 1: 7}
Dấu . , : ∗ # ’ ’ và ( ) [ ] { }

- sympy
  - binomial
  - coeff
  - doit
  - expand
  - factor
  - factorial
  - Sum
  - simplify
  - symbols

- itertools
  - combinations
  - combinations_with_replacement
  - permutations

Bài tập bổ sung

1.59. Xét khai triển đầy đủ của $\left(\frac{x}{2} + y - 3z\right)^5$.

a) Tìm hệ số của x^2yz^2.

b) Có bao nhiêu số hạng trong khai triển này?

c) Tìm tổng các hệ số trong khai triển.

1.60. Tìm số nghiệm nguyên không âm thỏa mãn:

a) $x_1 + x_2 + x_3 = 6$, $x_1 + x_2 + \cdots + x_5 = 15$.

b) $x_1 + x_2 + x_3 \leq 6$, $x_1 + x_2 + \cdots + x_5 \leq 15$.

1.61. Có bao nhiêu cách xếp các chữ cái của từ UNUSUAL? Trong số đó, có bao nhiêu cách xếp sao cho:

a) Tất cả chữ U kề nhau?

b) Không có hai chữ U nào kề nhau?

1.62. Tìm giá trị của biến `counter` sau khi thực hiện đoạn chương trình. Ở đây các biến nguyên $r \geq 1$, $s \geq 5$, và $t \geq 7$ đã có trước đoạn này.

```
counter := 10
for i := 1 to 12 do
```

```
    for j := 1 to r do
        counter := counter + 2
for k := 5 to s do
    for 1 := 3 to k do
        counter := counter + 4
for m := 3 to 12 do
    counter := counter + 6
for n := t downto 7 do
    counter := counter + 8
```

Hướng dẫn, đáp số

1.1 a) 97

```
from sympy import *

i = symbols('i')
Sum(i**2 + 1, (i, 1, 6)).doit()
```

b) -5 c) 12 d) 3

e) 0

```
n = symbols('n', odd=True)
Sum((-1)**k, (k, n, 2*n)).doit()
```

1.2 a) $\sum_{k=2}^{n} \frac{1}{k!}$ c) $\sum_{k=1}^{7} k^2$ e) $\sum_{k=0}^{n} (-1)^k \frac{n+k}{(2k)!}$

b) $\sum_{k=1}^{7} (-1)^{k-1} k^3$ d) $\sum_{k=0}^{n} \frac{k+1}{n+k}$

1.3 a) 1024 b) 59 049

1.4 Cách (2). Phần tử nào ràng buộc khó hơn thì chọn trước.

1.5 48, quy tắc cộng

1.6 576, quy tắc nhân

1.7 145

1.8

```
for a in range(1, 10):
    for b in range(10):
        for c in range(10):
            if c % 2 == 0 and a != b and a != c and b != c:
                print(a, b, c)
```

1.10

a) 40 320 b) 5 040 c) 720

1.11 60

1.12 24

1.13 a) 56

1.14 Xem công thức của $P(n, r)$ ở trang 18.

1.15 a) 10 b) 5 c) 5

```
from sympy import *
P = lambda n, r: factorial(n) / factorial(n-r)
n = symbols('n')

solve((P(n, 2) - 90).simplify(), n)  # [-9, 10]
```

1.16 a) FIT FTI IFT ITF TFI TIF

b) CEHU CEUH CHEU CHUE CUEH CUHE, ECHU ECUH EHCU EHUC EUCH EUHC, HCEU HCUE HECU HEUC HUCE HUEC, UCEH UCHE UECH UEHC UHCE UHEC

1.17

```
def chinh_hop_lap(a, r):
    if r == 1:
        return [[i] for i in a]
    A = []
    for p in chinh_hop_lap(a, r-1):
        for x in a:
            A.append(p + [x])
    return A

chinh_hop_lap([1, 2], 3)  # [[1, 1, 1], [1, 1, 2], [1, 2, 1], [1, 2, 2], [2, 1, 1], [2, 1, 2], [2, 2, 1], [2, 2, 2]]
```

1.19

$$\begin{aligned}\binom{n-1}{r} + \binom{n-1}{r-1} &= \frac{(n-1)!}{r!\,[(n-1)-r)]!} + \frac{(n-1)!}{(r-1)!\,[(n-1)-(r-1)!} = \frac{(n-1)!}{r!\,(n-r-1)!} + \frac{(n-1)!}{(r-1)!\,(n-r)!}\\ &= \frac{(n-1)!}{r!\,(n-r)!}[(n-r)+r] = \frac{(n-1)!}{r!\,(n-r)!} \times n = \frac{n!}{r!\,(n-r)!} = \binom{n}{r}\end{aligned}$$

1.20 a) Có $\binom{6}{2} = 15$ tổ hợp: ab, ac, ad, ae, af, bc, bd, be, bf, cd, ce, cf, de, df, ef.

b) Có $\binom{5}{3} = 10$ tổ hợp: afm, afr, aft, amr, amt, art, fmr, fmt, frt, mrt.

1.21 a) 792 b) 91 c) 3003

1.22 $(n-1)^2$

1.23

a) 28 b) 70 c) 28 d) 37

1.24 $\binom{n}{3}$, $\binom{n}{3} - n - n(n-4)$

1.25 a) $4 \times 4 \times 6$ b) bvx, egu

1.26 a) 220 b) 1 760 c) -3041280

1.27 a) $\sum_{i=0}^{n} \frac{1}{i!\,(n-i)!} = \frac{1}{n!} \sum_{i=0}^{n} \frac{n!}{i!\,(n-i)!} = \frac{1}{n!} \sum_{i=0}^{n} \binom{n}{i} = \frac{2^n}{n!}$.

b) $\sum_{i=0}^{n} \frac{(-1)^i}{i!\,(n-i)!} = \frac{1}{n!} \sum_{i=0}^{n} \frac{n!}{i!\,(n-i)!}(-1)^i = \frac{1}{n!} \sum_{i=0}^{n} \binom{n}{i}(-1)^i = \frac{1}{n!}0 = 0$.

1.28 $(m+1)\binom{m+n}{m+1} = (m+1)\frac{(m+n)!}{(m+1)!\,(m+n-m-1)!} = \frac{(m+n)!}{m!\,(n-1)!} = \frac{(m+n)!}{m!\,n!}n = n\binom{m+n}{m}$.

1.29 Tổng đã cho là khai triển của $(1+2)^n = 3^n$.

1.30 $(1+8)^{50} = x^{100} \Leftrightarrow x = \pm 3$

1.31 Áp dụng hằng đẳng thức Vandemonde với $m = r = n$, và tính đối xứng của các hệ số nhị thức đồng bậc.

1.32 Có 12 hoán vị.

1.33 a) 30 240 c) $4! \times 5 \times \binom{5}{3, 2}$

b) 2520 d) $\binom{6}{1, 1, 1, 2, 1} \times \binom{7}{3}$

1.34 a) 6 720 b) 720

1.35 a) 9 979 200

b) $\binom{10}{2, 2, 2, 3, 1} \times 22 \times 2! = 1\,663\,200$ c) $\binom{6}{2, 1, 2, 1} \times 7 \times \binom{6}{1, 2, 3} = 75\,600$

1.36 $\binom{7}{4, 1, 2} \times \binom{8}{4} = 7\,350$

1.37 $\binom{6}{1, 2, 1, 1, 1} + 2 \times \binom{6}{1, 2, 2, 1} = 720$

1.38 3 432

1.39 a–b) 360

c) Để đi từ điểm (a, b, c) đến điểm (x, y, z), với $a \le x, b \le y, c \le z$, ta có $\binom{dx + dy + dz}{dx, dy, dz}$ cách, trong đó $dx = x - a,\ dy = y - b,\ dz = z - c$.

1.41 a) 12 b) 12 c) -24 d) -216 e) 161 280

1.42

a) 113 400 b) 718 502 400

c) $124\,740\,000v^4 + 1\,496\,880\,000v^3 + 6\,735\,960\,000v^2 + 13\,471\,920\,000v + 10\,103\,940\,000$

1.43 a) 2^3 b) 2^{10} c) 3^{10} d) 4^5 e) 4^{10}

1.44 a) $\binom{4+32-1}{32}$ c) $\binom{4+8-1}{8}$ e) $\binom{4+40-1}{40}$

b) $\binom{4+28-1}{28}$ d) 1

f) $\binom{4+28-1}{28} - \binom{4+3-1}{3}$, trong đó số trừ là số các nghiệm thỏa mãn $x_4 \geq 26$

1.45 a) $\binom{6+39-1}{39}$ b) $\binom{6+54-1}{54}$

1.46 (1) có $\binom{19+(n-19)-1}{n-19}$ nghiệm, (2) có $\binom{64+(n-64)-1}{n-64}$ nghiệm. Từ $\binom{n-1}{18} = \binom{n-1}{63}$, suy ra $n-1 = 18+63$, nên $n = 82$.

1.47 a) $\binom{3+6-1}{6} \times \binom{4+31-1}{31}$ b) $\binom{5}{3}\binom{34}{31}$

1.48 a) $\binom{8}{2,4,1,0,1}3^2 2^4 = 120\,960$ b) $\binom{5+8-1}{8}$

1.49 $\binom{20+4-1}{4}$

1.50 $10 + \binom{15+3-1}{3}$

1.51 a) **Cách 1:** Xếp vào mỗi hộp một vật. Sau đó xếp $m-n$ vật vào n hộp. Số cách là $\binom{n+(m-n)-1}{m-n}$.

Cách 2: Đặt x_i là số vật ở hộp i, $1 \leq i \leq n$. Mỗi cách xếp như vậy tương ứng với một nghiệm nguyên dương của phương trình $x_1 + x_2 + \cdots + x_n = m$.

1.52 a)

```
for x1 in range(11):
    for x2 in range(11):
        for x3 in range(11):
            if x1 + x2 + x3 == 10:
                print(x1, x2, x3)
```

hoặc

```
for x1 in range(11):
    for x2 in range(11 - x1):
        x3 = 10 - x1 - x2
        print(x1, x2, x3)
```

b) Đặt $y_i = x_i + 2$, $1 \leq i \leq 4$, thì $y_i \geq 0$, và $y_1 + y_2 + y_3 + y_4 = 12$.

```
for y1 in range(13):
    for y2 in range(13 - y1):
        for y3 in range(13 - y1 - y2):
            y4 = 12 - y1 - y2 - y3
            x1, x2, x3, x4 = y1 - 2, y2 - 2, y3 - 2, y4 - 2
            print(x1, x2, x3, x4)
```

1.53

```
def solutions(n, r):
    if n==1:
        return [[r]]
    if r==0:
        return [[0 for _ in range(n)]]
    S = []
    for s in solutions(n-1, r):
        s = [0] + s
        S.append(s)
    for s in solutions(n, r-1):
        s[0] += 1
        S.append(s)
    return S
```

1.54 14532, 15432, 21345, 23451, 23514, 31452, 31542, 43521, 45213, 45321.

1.55 156423, 165432, 231456, 231465, 234561, 314562, 432561, 435612, 541236, 543216, 654312, 654321

1.56 a) 2134 c) 12534 e) 6714253

b) 54132 d) 45312 f) 31542678

1.57

```
a = [3, 6, 2, 5, 4, 1]
while a is not None:
    print(a)
    a = next_permutations(a)
```

1.58

```
a = [1, 2, 5, 6]
n = 6
r = len(a)

while a is not None:
    print(a)
    a = next_combinations(n, a)
```

1.59 a) $\binom{5}{2,1,2}\left(\frac{1}{2}\right)^2(-3)^2 = \frac{135}{2}$ b) $\binom{3+5-1}{5}$

c) Thay x, y, z bằng 1

1.60 a) $x_1 + x_2 + x_3 = 5$, $x_4 + x_5 = 9$ có $\binom{3+6-1}{6}\binom{2+9-1}{9}$ nghiệm.

b) Đặt $x_1+x_2+x_3 = k, 0 \le k \le 6$, có $\binom{3+k-1}{k}$ nghiệm x_1, x_2, x_3. Khi đó $x_4+x_5 \le 15-k$, hay $x_4+x_5+x_6 = 15-k$, trong đó $x_4, x_5, x_6 \ge 0$ có $\binom{3+(15-k)-1}{15-k}$ nghiệm x_4, x_5, x_6. Số nghiệm cần tìm $\sum_{k=0}^{6}\binom{k+2}{k}\binom{17-k}{15-k}$.

1.61

$\binom{7}{1,1,1,1,3}$.

a) $5!$

b) $\binom{5}{3}4!$

1.62 $10 + 12 \cdot r \cdot 2 + 4\sum_{k=5}^{s}(k-2) + 10 \cdot 6 + 8(t-6) = 24r + 2s^2 - 6s + 8t + 14$

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*. phiên bản 6. Pearson, 2022. 755 trang.

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

[4] Ralph P. Grimaldi. *Discrete and Combinatorial Mathematics: Instructor's Solutions Manual*. phiên bản 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*. phiên bản 8. McGraw-Hill Education, 2019. 1118 trang.

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

[8] Watson S. Stewart J. Clegg D. *Calculus: Early Transcendentals*. phiên bản 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.