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: phương án một có m cách thực hiện, phương án hai có n cách thực hiện, và không thể thực hiện đồng thời hai phương án, thì có $m + n$ cách thực hiện công việc.

Ví dụ 1.1. Trong thư viện ở một trường đại học, các giáo trình về toán ứng dụng gồm 40 giáo trình của nhà xuất bản A, và 50 giáo trình về toán ứng dụng của nhà xuất bản B. Có bao nhiêu cách để sinh viên chọn một giáo trình về toán ứng dụng?

Giải. Sinh viên có $40 + 50 = 90$ cách chọn một giáo trình về toán ứng dụng. □

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.

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. Sinh viên có $5 + 5 + 5 = 15$ cách chọn một sách để học. □

Ví dụ 1.3. Với số nguyên dương n, xét đoạn chương trình

```
1 counter = 0
2 for i = 1,n
3     for j = 1,i
4         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)

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

print(counter)
```

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

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

Vì vậy, giá trị sau cùng của `counter` bằng số chu trình.

Với mỗi $i = 1, 2, \dots, n$, dòng 3–4 duyệt i cặp (i, j) được duyệt. Theo quy tắc cộng, số chu trình là

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

□

Tính tổng trong ví dụ trên bằng lệnh sau

```python
from sympy import *
n, i = symbols('n i')
Sum(i, (i, 1, n)).doit().simplify()
```

Quy tắc nhân: Nếu có hai bước thực hiện một công việc: bước một có m cách thực hiện, và 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.

Ví dụ 1.4. Câu lạc bộ kịch của trường đại học tổ chức buổi thử vai cho một vở diễn. Có sáu nam và tám nữ thử vai nam và nữ chính.
Đạo diễn có bao nhiêu cách chọn cặp diễn viên chính?

Giải. Số cách chọn cặp diễn viên chính là $6 \times 8 = 48$. □

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

Ví dụ 1.5. a) Với số nguyên dương n, có bao nhiêu xâu nhị phân độ dài n?

b) Với $n \geq 2$, tìm một quy tắc xây dựng các xâu nhị phân độ dài n từ các xâu nhị phân độ dài $n-1$. Lấy một ví dụ cụ thể với $n = 2, 3$.

Giải. a) Mỗi xâu nhị phân độ dài n có dạng $\underbrace{*\ *\ *\cdots *\ *}_{n}$, trong đó mỗi dấu $*$ có 2 cách chọn là 0 hoặc 1. Theo quy tắc nhân, số xâu nhị phân là

$$\underbrace{2 \times 2 \times \cdots \times 2}_{n} = 2^n.$$

b) Ứng với mỗi xâu độ dài $n-1$, ta lập xâu độ dài n bằng cách xếp 0 hoặc 1 vào đầu xâu đó.

Có hai xâu độ dài 1 là 0 và 1. Nếu xếp 0 vào đầu hai xâu này được 00, 01. nếu xếp 1 vào đầu, được 10, 11. Ta được 4 xâu độ dài 2 là 00, 01, 10, 11.

Tiếp theo, xếp 0 vào đầu các xâu độ dài 2, được 000, 001, 010, 011. Nếu xếp 1 vào đầu các xâu đó, được 100, 101, 110, 111. Ta được 8 xâu độ dài 3, là 000, 001, 010, 011, 100, 101, 110, và 111.

□

Trong Python, để liệt kê xâu nhị phân độ dài 3

```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)]
```

Để lưu dữ liệu, bộ nhớ máy tính chứa một lượng lớn các mạch, mỗi mạch có khả năng lưu trữ một *bit*, tức là một trong hai mã nhị phân 0 hoặc 1. Các mạch lưu trữ này được sắp xếp trong các khối gọi là ô nhớ. Để xác định các ô trong bộ nhớ, mỗi ô được gán một tên duy nhất gọi là *địa chỉ* của nó. Với một số máy tính, chẳng hạn như bộ vi điều khiển nhúng (ví dụ, trong hệ thống đánh lửa cho ô tô), một địa chỉ được biểu diễn bằng dãy tám bit, gọi là *byte*. Có $\underbrace{2 \times 2 \times \cdots \times 2}_{8} = 2^8 = 256$ byte như vậy, tức là có 256 địa chỉ có thể để sử dụng cho các ô nhớ.

Một thiết bị nhà bếp, chẳng hạn như lò vi sóng, tích hợp một bộ vi điều khiển nhúng. Những "máy tính nhỏ" này (chẳng hạn như vi điều khiển PICmicro) chứa hàng nghìn ô nhớ và sử dụng địa chỉ hai byte để xác định các ô này trong bộ nhớ chính của chúng. Các địa chỉ như vậy được tạo thành từ hai byte liên tiếp, hay 16 bit liên tiếp. Như vậy có $256 \times 256 = 2^8 \times 2^8 = 2^{16} = 65\,536$ địa chỉ có thể, được sử dụng để xác định các ô trong bộ nhớ. Các máy tính khác dùng hệ thống địa chỉ bốn byte. Kiến trúc 32-bit này được sử dụng trong bộ xử lý Pentium[*], với $2^8 \times 2^8 \times 2^8 \times 2^8 = 2^{32} = 4\,294\,967\,296$ địa chỉ để xác định các ô trong bộ nhớ. Bộ xử lý UltraSPARC[†] hoặc Itanium[‡] cung cấp địa chỉ tám byte, mỗi địa chỉ bao gồm $8 \times 8 = 64$ bit, và có thể có $2^{64} = 18\,446\,744\,073\,709\,551\,616$ địa chỉ cho kiến trúc này. (Tất nhiên, không phải tất cả các khả năng này đều thực sự được sử dụng.)

Ví dụ 1.6. 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) Số các biển xe mã vùng Hà Nội là

$$6 \times 26 \times 10 \times \underbrace{10 \times 10 \times \cdots \times 10}_{5} = 156 \times 10^6.$$

```python
6 * 26 * 10 * 10**5
```

[*] Pentium (R) là nhãn hiệu của Intel

[†] Bộ xử lý UltraSPARC do Sun (R) Micorsystem sản xuất

[‡] Itanium là nhãn hiệu của Intel Corporation

b) 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 1.7. Với số nguyên dương m, n, xét đoạn chương trình

```
1 for i = 1,m
2     for j = 1,n
3         print i, j
```

a) Viết chương trình bằng Python với $m = 4$, $n = 3$. 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
counter = 0
for i in range(1, m+1):
    for j in range(1, n+1):
        counter += 1
        print(counter, i, j)
```

b) Ứng với mỗi $i = \overline{1, m}$, dòng 2–3 thực hiện n lệnh `print`. Do đó, chương trình thực hiện $m \times n$ lệnh `print`.

□

Ví dụ 1.8. Cho giả mã của thuật toán nhân ma trận $A = (a_{ij})_{m\times n}$ với $B = (b_{ij})_{n\times p}$, cho ta ma trận $C = (c_{ij})_{m\times p}$ xác định bởi $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$.

```
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
```

Thuật toán thực hiện bao nhiêu phép toán số học?

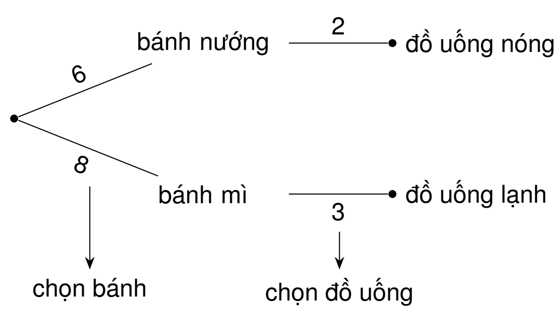
Giải. Dòng 5 thực hiện 2 phép toán số học. Theo quy tắc nhân, dòng 4–5 thực hiện $n \times 2 = 2n$ phép toán. Tương tự 2–5 thực hiện $p \times 2n = 2np$ phép toán, và do đó, thuật toán thực hiện $m \times 2np = 2mnp$ phép toán. □

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 bước sau thay đổi phụ thuộc kết quả của bước trước.

Ví dụ 1.9. Tại một cửa hàng phục vụ sáu loại bánh nướng, tám loại bánh mì và năm 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. Vì vậy 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.

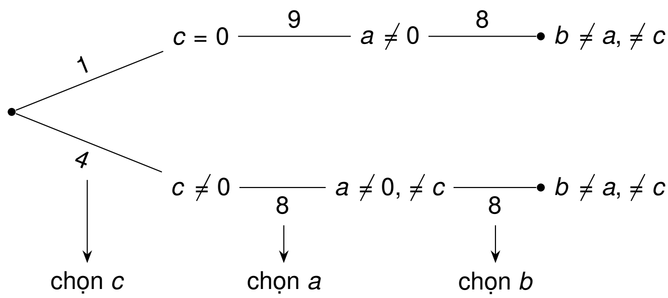
Ví dụ 1.10. Mô tả cách đếm các số tự nhiên chẵn có ba chữ số khác nhau.

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: $\neq 0, \neq b, \neq c$ $\quad$ b: $\neq a, \neq c$, và $\quad$ 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

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

□

Bài tập 1.1

1.1. Trong Ví dụ 1.10, 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.

1.2. 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
```

1.3. 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?

1.4. 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?

1.5. 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.

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

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

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

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

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

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

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

1.8. Lập trình bài toán nhân ma trận. Kiểm tra với

$$\begin{bmatrix} 2 & 0 & -1 \\ 1 & 3 & -2 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 1 & 0 \\ 2 & 3 & -1 & 4 \\ -3 & 0 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 3 & -2 & 4 & -1 \\ 12 & 8 & 2 & 10 \end{bmatrix}$$

1.9. Trong Ví dụ 1.10, lập trình liệt kê các số tự nhiên chẵn có ba chữ số khác nhau.

1.10. * Cho số nguyên dương n. Áp dụng ý tưởng ở Ví dụ 1.5(b), lập trình liệt kê các xâu nhị phân độ dài n.

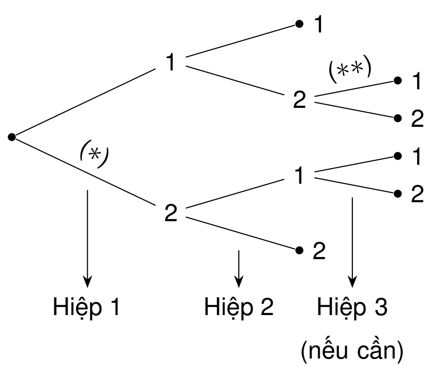
1.2 Biểu đồ cây

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 ba hiệp, biết rằng trận đấu kết thúc khi có tay vợt vừa thắng hai hiệp, và đó là người thắng trận.

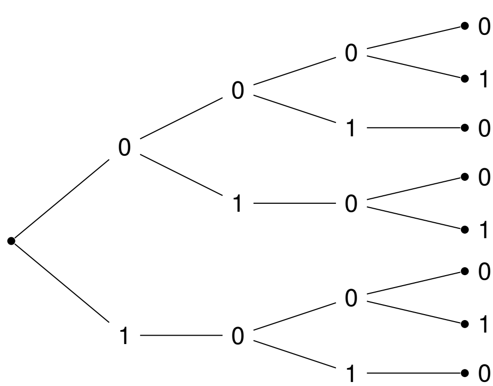
Giải. Đánh số hai tay vợt là 1 và 2. Mỗi hiệp ứng với một tầng của cây, điểm bên phải của mỗi nhánh tại tầng đó đánh dấu người thắng trong hiệp này. Chẳng hạn, nhánh $*$ cho biết người 2 thắng hiệp 1, nhánh $(**)$ ngụ ý người 1 thắng trận đấu vì thắng hiệp 1 và hiệp 3.



□

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.



□

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*.

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

Định nghĩa 1.2. *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 $a_1, a_2, \ldots, a_n$ là một bộ có thứ tự $(a_{i_1}, a_{i_2}, \ldots, a_{i_r})$. Nếu không gây nhầm lẫn, có thể viết $a_{i_1} a_{i_2} \ldots a_{i_r}$.

Xâu nhị phân độ dài n là chỉnh hợp lặp chập n của 0 và 1.

Định nghĩa 1.3. *Với số tự nhiên n, n giai thừa, ký hiệu n!, xác định bởi*

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

Số hoán vị của n vật phân biệt là

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

Số hoán vị lặp 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 1.13. a) Các số 1, 2, 3 có 3! = 6 hoán vị

123	132	213	231	312 và	321

```python
from sympy import *
factorial(3)

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

b) Số chỉnh hợp chập 3 của 1, 2, 3, 4 là $P(4, 3) = \dfrac{4!}{(4-3)!} = 24$, gồm

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

```python
from sympy import *
factorial(4) / factorial(4-3)

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

c) Các chữ cái trong từ COMPUTER có 8! = 40 320 hoán vị, và số từ có 5 chữ cái lấy ra từ nó là $P(8, 5) = \dfrac{8!}{(8-5)!} = 6\,720$ từ.

d) Số cách chọn 4 người trong 10 người và xếp thành hàng để chụp ảnh là $P(10, 4) = \dfrac{10!}{(10-4)!} = 5\,040$.

Ví dụ 1.14. Nêu quy tắc xây dựng các hoán vị của các số $1, 2, \ldots, n$ từ các hoán vị của n số. Áp dụng cho $n = 3, 4$.

Giải. Với $n \geq 2$, ta lập hoán vị của $a_1, a_2, \ldots, a_n$, bằng cách xếp a_i vào đầu các hoán vị của dãy con $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n$.

$a_1, \ldots, a_n$	a_i	**Dãy con**	**Hoán vị của dãy con**	**Hoán vị của** $a_1, a_2, \ldots, a_n$

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

1, 2	**1**	2	2	**1**2
	2	1	1	**2**1
1, 2, 3	**1**	2, 3	23, 32	**1**23, **1**32
	2	1, 3	13, 31	**2**13, **2**31
	3	1, 2	12, 21	**3**12, **3**21
1, 2, 3, 4	1	2, 3, 4	234, 243, 324, 342, 423, 423	1234, 1243, 1324, 1342, 1423, 1423
	2	1, 3, 4	134, 143, 314, 341, 413, 431	2134, 2143, 2314, 2341, 2413, 2431
	3	1, 2, 4	124, 142, 214, 241, 412, 421	3124, 3142, 3214, 3241, 3412, 3421
	4	1, 2, 3	123, 132, 213, 231, 312, 321	4123, 4132, 4213, 4231, 4312, 4321

□

Ví dụ 1.15. Cho số nguyên dương n, r, với $r \geq 2$. Nêu một quy tắc lập các chỉnh hợp lặp chập r của $1, 2, \ldots, n$ từ các chỉnh hợp chập $r-1$ của n số đó. Minh họa với $n = 4$ và $r = 2, 3$.

Giải. Mỗi chỉnh hợp lặp chập r của $1, 2, \ldots, n$ có dạng $a_{i_1} a_{i_2} \ldots a_{i_{r-1}} a_{i_r}$ trong đó $a_{i_1} a_{i_2} \ldots a_{i_{r-1}}$ là chỉnh hợp lặp chập $r-1$ của n số đó, và $a_{i_r} = 1, 2, \ldots, n$.

Chẳng hạn, với các chỉnh hợp lặp chập 1, ta lập được các chỉnh hợp lặp chập 2. Từ chỉnh hợp lặp chập 2, ta lại lập tiếp các chỉnh hợp lặp chập 3.

r	$a_{i_1} a_{i_2} \ldots a_{i_{r-1}}$	$a_{i_1} a_{i_2} \ldots a_{i_{r-1}} a_{i_r}$
2	1	11, 12, 13, 14
	2	21, 22, 23, 24
	3	31, 32, 33, 34
	4	41, 42, 43, 44
3	11	111, 112, 113, 114
	12	121, 121, 123, 124
	13	131, 132, 133, 134
	...	...
	44	441, 442, 443, 444

□

Ví dụ 1.16. Cho số nguyên dương $n \geq r \geq 2$. Nêu một quy tắc lập các chỉnh hợp chập r của $1, 2, \ldots, n$ từ các chỉnh hợp chập $r-1$ của n số đó. Minh họa với $n = 4$ và $r = 2, 3$.

Giải. Mỗi chỉnh hợp chập r của $1, 2, \ldots, n$ có dạng $a_{i_1} a_{i_2} \ldots a_{i_{r-1}} a_{i_r}$ trong đó $a_{i_1} a_{i_2} \ldots a_{i_{r-1}}$ là chỉnh hợp chập $r-1$ của n số đó, và a_{i_r} khác các số $a_{i_1}, a_{i_2}, \ldots, a_{i_{r-1}}$.

Chẳng hạn, với các chỉnh hợp chập 1, ta lập được các chỉnh hợp chập 2. Từ chỉnh hợp chập 2, ta lại lập tiếp các chỉnh hợp chập 3.

r	$a_{i_1} a_{i_2} \ldots a_{i_{r-1}}$	a_{i_r}	$a_{i_1} a_{i_2} \ldots a_{i_{r-1}} a_{i_r}$
2	1	2, 3, 4	12, 13, 14
	2	1, 3, 4	21, 23, 24
	3	1, 2, 4	31, 32, 34
	4	1, 2, 3	41, 42, 43
3	12	3, 4	123, 124
	13	2, 4	132, 134
	14	2, 3	142, 143
	...	...	...
	43	1, 2	431, 432

□

Bài tập 1.3

1.11. Liệt kê các hoán vị của

a) FIT

b) HUCE

1.12. 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?

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

1.14. Tính

a) $P(7,2)$

b) $P(8,4)$

c) $P(10,7)$

d) $P(12,3)$

1.15. 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?

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

1.17. Tìm n thỏa mãn

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

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

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

1.18. Cho số tự nhiên n. Lập trình tính $n!$. [*Gợi ý:* đệ quy hoặc dùng vòng lặp.]

1.19. * Cho số tự nhiên $n \geq 1$. Áp dụng ý tưởng của Ví dụ 1.14, lập trình tìm các hoán vị của các số $1, 2, \ldots n$. [*Gợi ý:* đệ quy.]

1.20. * 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*]

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

1.22. * Lập trình liệt kê tất cả biển số xe trong hai tình huống ở Ví dụ 1.6.

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 $n \geq r \geq 0$.

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

Ví dụ 1.17. Liệt kê và đếm các tổ hợp chập 3 của 5 số 1, 2, 3, 4, 5.

Giải. Có $\binom{5}{3} = 10$ tổ hợp chập 3 của 5 số này, gồm

123	125	135	234	245
124	134	145	235	345

□

```python
from sympy import *
binomial(5, 3)

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

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

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

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

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

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 $* * * \ldots * *$, trong đó mỗi dấu $*$ lần lượt là số hạng x hoặc y trong các dấu ngoặc. Suy ra mỗi số hạng có dạng $x^r y^s$.

* 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

Mặt khác, thừa số tại mỗi dấu $*$ là x hoặc y, mà có n thừa số, nên $r + s = n$, tức là số hạng có dạng $x^r y^{n-r}$.

Cuối cùng, ta rút gọn các số hạng đồng dạng, tức là với mỗi r, $0 \leq r \leq n$, 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; và (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. □

Python chỉ khai triển được nhị thức với n cụ thể

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

nhưng có thể rút gọn vế phải trong trường hợp cụ thể

```
( x**2 + 2 * x * y + y**2 ).refine()   # x^2 + 2xy + y^2 = (x + y)^2
```

hoặc tổng quát

```
n = symbols('n')
Sum(binomial(n, r) * x**r * y**(n-r), (r, 0, n)).doit().
    simplify()
```

mặc dù kết quả khá phức tạp:

$$\begin{cases} y^n \left(\frac{x+y}{y}\right)^n & \text{for } \left(\operatorname{re}(n) \leq -1 \vee \left|\frac{x}{y}\right| \leq 1\right) \wedge \left(\left|\frac{x}{y}\right| \leq 1 \vee \left|\frac{x}{y}\right| < 1\right) \wedge \cdots \\ y^n \sum_{r=0}^{n} x^r y^{-r} \binom{n}{r} & \text{otherwise} \end{cases}$$

Định lý nhị thức cho ta khẳng định về tổng các hệ số nhị thức.

Hệ quả 1.1. *Với số tự nhiên n,*

a) $\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n} = 2^n$, *và*

b) $\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \cdots + (-1)^n \binom{n}{n} = 0.$

Chứng minh. a) **Cách 1:** (phép đếm) Trong phần chứng minh của Định lý nhị thức, số số hạng có dạng $x^r y^{n-r}$ là $\binom{n}{r}$, với $0 \leq r \leq n$. Theo quy tắc cộng, số số hạng trong khai triển là $\sum_{r=0}^{n} \binom{n}{r}$.

Mặt khác, mỗi số hạng ứng với một cách chọn n thừa số, là x hoặc y, lần lượt trong n biểu thức $x + y$. Vì mỗi thừa số có hai cách chọn, theo quy tắc nhân, số số hạng lập được là $\underbrace{2 \cdot 2 \cdot 2 \cdots 2}_{n} = 2^n$. Vậy $\sum_{r=0}^{n} \binom{n}{r} = 2^n$.

Cách 2: (biểu thức toán). Trong khai triển $(x + y)^n = \sum_{r=0}^{n} \binom{n}{r} x^r y^{n-r}$, cho $x = y = 1$, ta có $(1 + 1)^n = \sum_{r=0}^{n} \binom{n}{r} 1^r 1^{n-r}$, và rút gọn là xong!

```
from sympy import *
n, r = symbols('n r')
Sum(binomial(n, r), (r, 0, n)).doit()
```

b) Trong khai triển $(x + y)^n = \sum_{r=0}^{n} \binom{n}{r} x^{n-r} y^r$, cho $x = 1, y = -1$.

```
Sum((-1)**r * binomial(n, r), (r, 0, n)).doit()
```

□

Định lý 1.2 (Hằng đẳng thức Pascal). * *Cho số nguyên $n \geq r \geq 1$. Chứng minh*

$$\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}. \tag{1.6}$$

Giải. **Cách 1:** Xét các cách chọn r vật từ n vật $a_1, a_2, \dots, a_{n-1}, x$. Trong $\binom{n}{r}$ cách chọn này, có hai khả năng:

1) Không chọn được x. Khi đó cách chọn này tương ứng với cách chọn r vật từ $n - 1$ vật $a_1, a_2, \dots, a_{n-1}$, và số cách là $\binom{n-1}{r}$.
2) Chọn được x. Cách chọn này tương ứng với cách chọn $r - 1$ vật còn lại trong $n - 1$ vật $a_1, a_2, \dots, a_{n-1}$, và số cách là $\binom{n-1}{r-1}$.

Theo quy tắc cộng, ta có $\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}$.

Cách 2:

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

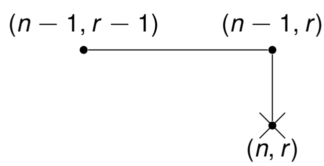
*Blaise Pascal, 1623–1662, nhà toán học, vật lý, triết học Pháp

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

□

```python
from sympy import *
n, r = symbols('n r', integer=True)
( binomial(n-1, r) + binomial(n-1, r-1) ).simplify()
```

Lược đồ mô tả hằng đẳng thức Pascal



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

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*

$$\binom{m+n}{r} = \sum_{k=0}^{r} \binom{m}{k}\binom{n}{r-k}. \tag{1.7}$$

Chứng minh. content... □

*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.

Ví dụ 1.18. Mô tả phương pháp liệt kê các tổ hợp chập r của $a_1, a_2, \dots, a_n$. Lấy ví dụ cho các tổ hợp chập 3 của 1, 2, 3, 4, 5.

Giải. **Cách 1:** trong chứng minh hằng đẳng thức Pascal bằng phép đếm ở Định lý 1.2, các tổ hợp chập r của $a_1, a_2, \dots, a_n$ gồm

1) các tổ hợp chập r của $a_2, a_3, \dots, a_n$; và
2) các tổ hợp cấu thành từ tổ hợp chập $r-1$ của $a_2, a_3, \dots, a_n$ bằng cách bổ sung a_1 vào.

Với $r = 1$, các tổ hợp chập 1 của $a_1, a_2, \dots, a_n$ là các tổ hợp $\{a_1\}, \{a_2\}, \dots, \{a_n\}$. Với $r = n$, chỉ có một tổ hợp chập n của $1, 2, \dots, n$ là tổ hợp $a_1 a_2 \dots a_n$.

Chẳng hạn, tổ hợp chập 3 của 1, 2, 3, 4, 5 bao gồm

1) tổ hợp chập 3 của 2, 3, 4, 5, gồm
   i) tổ hợp chập 3 của 3, 4, 5, là 345; và
   ii) tổ hợp chập 2 của 3, 4, 5, và ghép với 2, gồm **2**34, **2**35, **2**45
2) tổ hợp chập 2 của 2, 3, 4, 5, và ghép với 1, gồm **1**12, **1**24, **1**25, **1**34, **1**35, **1**45.

Cách 2: Mỗi tổ hợp chập r của $a_1, a_2, \dots, a_n$ là một dãy $a_{i_1} a_{i_2} \dots a_{i_r}$ thỏa mãn

$$1 \le i_1 < i_2 < \dots < i_r \le n.$$

Trước hết, nhận xét $i_{r-1} \le n-1, i_{r-2} \le n-2, \dots, i_1 \le n-(r-1) = n-r+1$. Đặt $i_1 = k$, thì $1 \le k \le n-r+1$, và $i_2 > k$ hay $i_2 \ge k+1$.

Với mỗi $k = 1, 2, \dots, n-r+1$, ta có $k+1 \le i_2 < i_3 < \dots < i_r \le n$ ứng với tổ hợp $a_{i_2} a_{i_3} \dots a_{i_r}$ chập $r-1$ của $a_{k+1}, a_{k+2}, \dots, a_n$.

Với $r = 1$, các tổ hợp chập 1 của $a_1, a_2, \dots, a_n$ là các tổ hợp $\{a_1\}, \{a_2\}, \dots, \{a_n\}$.

Chẳng hạn, tổ hợp chập 3 của 1, 2, 3, 4, 5 gồm

k	a_k	$a_{k+1}, \dots, a_n$	tổ hợp chập 2 của $a_1, \dots, a_{k-1}$	tổ hợp chập 3 của $a_1, \dots, a_n$
1	1	2, 3, 4, 5	23, 24, 25, 34, 35, 45	**1**23, **1**24, **1**25, **1**34, **1**35, **1**45
2	2	3, 4, 5	34, 35, 45	**2**34, **2**35, **2**45
3	3	4, 5	45	**3**45

□

Bài tập 1.4

1.23. 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.

1.24. Tính

a) $\binom{10}{4}$ b) $C(12,7)$ c) C_{14}^{12} d) $\binom{15}{10}$

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

1.26. Có bao nhiêu byte có

a) hai số 1? b) bốn số 1? c) sáu số 1? d) ít nhất sáu số 1?

1.27. 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.

1.28. 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

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

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

1.30. Cho số nguyên dương n. Tính

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

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

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

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

1.34. Với $n \in \mathbb{Z}^+$, chứng minh $\binom{2n}{n} = \sum_{k=0}^{n} \binom{n}{k}^2$.

1.35. Áp dụng công thức ở Định lý 1.2, lập trình tính $\binom{n}{r}$. [*Gợi ý:* đệ quy hoặc quy hoạch động (xem lược đồ ở phía sau định lý).]

1.36. Cho các số nguyên dương $n \geq r \geq 1$. Theo Ví dụ 1.18, lập trình tìm các tổ hợp chập r của $1, 2, \ldots, n$.

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 là

$$\frac{n!}{n_1!\, n_2! \cdots n_r!}. \tag{1.8}$$

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 một 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 $\dfrac{n!}{n_1!\,n_2!\cdots n_r!}$.

Định lý 1.4. *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*

$$\frac{n!}{n_1!\,n_2!\cdots n_r!},$$

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

Ký hiệu hệ số này bởi $\binom{n}{n_1, n_2, \dots, n_r}$, gọi là hệ số đa thức.

Chứng minh. □

Ví dụ 1.19. Từ BALL có $\dfrac{4!}{1!1!2!} = 12$ hoán vị lặp, gồm

1) A B L L	4) B A L L	7) B L A L	10) B L L A
2) A L B L	5) L A B L	8) L B A L	11) L B L A
3) A L L B	6) L A L B	9) L L A B	12) L L B A

Ví dụ 1.20. 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?

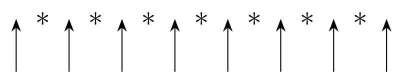
Giải. a) Số lượng mỗi chữ cái của từ MASSASAUGA là A: 4, G: 1, M: 1, S: 3, U: 1. Vậy từ có $\dfrac{10!}{4!\ 1!\ 1!\ 3!\ 1!} = 25\,200$ hoán vị.

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 $******$.

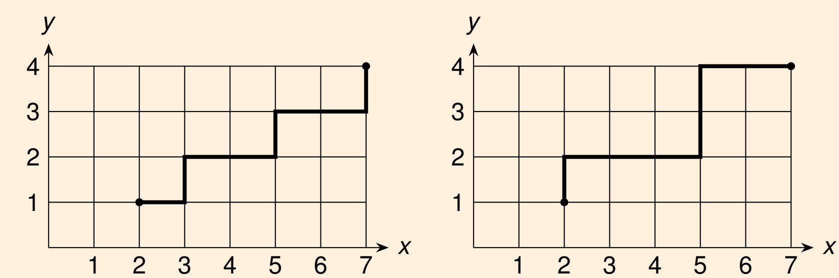
2) Với mỗi hoán vị trên, chèn bốn chữ A vào bốn vị trí đầu hoặc giữa, mỗi vị trí một chữ A, thì sẽ không có chữ A cạnh nhau. Số cách chèn bốn chữ A như vậy là số cách chọn 4 vị trí trong 7 vị trí có thể chèn, và bằng $\binom{7}{4} = 35$.



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

$\square$

Ví dụ 1.21. 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

Có bao nhiêu đường đi như vậy?

Giải. 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 $\dfrac{8!}{5!\,3!} = 56$. $\square$

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

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

Ví dụ 1.22. Khai triển $(x + y + z)^3$.

Giải.

n	Khai triển	Đơn thứ	Hệ số
3	$= 3 + 0 + 0$	$x^3,\ y^3,\ z^3$	$C_3^{3,0,0} = \dfrac{3!}{3!0!0!} = 1$
	$= 2 + 1 + 0$	$x^2y,\ x^2z,\ xy^2,\ y^2z,\ xz^2,\ yz^2$	$C_3^{2,1,0} = \dfrac{3!}{2!1!0!} = 3$
	$= 1 + 1 + 1$	xyz	$C_3^{1,1,1} = \dfrac{3!}{1!1!1!} = 6$

Ta được

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

□

Ví dụ 1.23. a) Trong khai triển của $(x + y + z)^7$, tìm hệ số của các đơn thức $x^2y^2z^3$, xyz^5, và x^3z^4.

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

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

Đơn thức	Hệ số
$x^2y^2z^3$	$\dbinom{7}{2,2,3} = \dfrac{7!}{2!\ 2!\ 3!} = 210$
xyz^5	$\dbinom{7}{1,1,5} = \dfrac{7!}{1!\ 1!\ 5!} = 42$
x^3z^4	$\dbinom{7}{3,0,4} = \dfrac{7!}{3!\ 0!\ 4!} = 35$

```
from sympy import *
x, y, z = symbols('x y z')
( (x + y + z)**7 ).expand().coeff(x**2 * y**2 * z**3)
```

b) Số hạng chứa a^4bc^3 trong khai triển của $(a-2b+3c+5)^{10}$ là $\dbinom{10}{4,1,3,2} a^4(-2b)(3c)^3 5^2$.

Hệ số cần tìm là $\dfrac{10!}{4!\ 1!\ 3!\ 2!}(-2)3^35^2 = -17\ 010\ 000$.

□

Bài tập 1.5

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

1.38. 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?

1.39. Từ DATAGRAM* có bao nhiêu

a) hoán vị?

b) hoán vị có tất cả chữ A cạnh nhau?

1.40. 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?

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

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

1.43. Có bao nhiêu cách đi trên lưới nguyên *Oxy* từ

a) $(0, 0)$ đến $(7, 7)$

b) $(2, 7)$ đến $(9, 14)$

1.44. Lập trình tìm các đường đi trên lưới nguyên *Oxy* từ điểm (a, b) đến điểm (x, y), với $a \leq x, b \leq y$.

1.45.

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

$$\begin{aligned} &(\text{H}) &&: \quad (x, y, z) \text{ đến } (x+1, y, z) \\ &(\text{V}) &&: \quad (x, y, z) \text{ đến } (x, y+1, z) \\ &(\text{A})^{*} &&: \quad (x, y, z) \text{ đến } (x, y, z+1) \end{aligned}$$

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).

1.46. * Lập trình liệt kê các đường đi trên lưới nguyên *Oxyz*, từ điểm (a, b, c) đến (x, y, z), với $x \leq a, y \leq b, z \leq c$. [*Gợi ý:* vận dụng ý tưởng của Bài 1.44.]

---

*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

1.47. 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 $(2w-x+3y-2z)^8$

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

a) $(w+x+y+z+1)^2$

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

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

1.49. 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

$$\binom{n+r-1}{r} = \frac{(n+r-1)!}{r!\,(n-1)!}. \tag{1.9}$$

Chứng minh. content... □

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

a) số nghiệm nguyên của phương trình

$$x_1 + x_2 + \cdots + x_n = r, \quad x_i \geq 0, 1 \leq i \leq n. \tag{1.10}$$

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

Ví dụ 1.24. Số cách chọn 4 sản phẩm từ 3 loại sản phẩm A, B, C là $\binom{3+4-1}{4} = 15$, gồm

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

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

Ví dụ 1.25. Trong đoạn chương trình sau, lệnh `print` thực hiện bao nhiêu lần

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

Giải. Đoạn chương trình duyệt qua các số nguyên i, j, k thỏa mãn $1 \leq k \leq j \leq i \leq 20$. Mỗi bộ (i, j, k) như vậy tương ứng 1-1 với tổ hợp lặp chập 3 của các số từ 1 đến 20. Do đó, lệnh `print` được thực hiện $\binom{20+3-1}{3} = 1\,540$ lần. □

Ví dụ 1.26. 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?

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

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

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

□

Ở trang 27, ta đã đếm các nghiệm nguyên không âm của phương trình $x_1 + x_2 + \cdots + x_n = r$. Bây giờ ta xét phương trình nghiệm nguyên này điều kiện tổng quát $x_i \geq a_i$, a_i là số nguyên, $i = \overline{1, n}$. Đặt $x_i = x_i' + a_i$, ta được phương trình nghiệm nguyên không âm tương đương

$$x_1' + x_2' + \cdots + x_n' = r' = r - a_1 - a_2 - \cdots - a_n,$$

với số nghiệm là $\binom{n+r'-1}{r'}$.

Ví dụ 1.27. 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) Mỗi nghiệm nguyên không âm của $x_1 + x_2 + \cdots + x_6 < 10$ tương ứng 1-1 với một nghiệm nguyên không âm của phương trình $(x_1 + x_2 + \cdots + x_6) + x_7 = 10$ với $x_7 \geq 1$. Số nghiệm cần tìm là $\binom{7+9-1}{9} = 5\,005$.

□

Ý (b) của Ví dụ 1.27 có thể giải theo cách sau. 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

$$\begin{aligned}
\sum_{r=0}^{9}\binom{6+r-1}{r} = \sum_{r=0}^{9}\binom{5+r}{r} &= \binom{5}{0}+\binom{6}{1}+\binom{7}{2}+\binom{8}{3}+\cdots+\binom{14}{9}\\
&= \left[\binom{6}{0}+\binom{6}{1}\right]+\binom{7}{2}+\binom{8}{3}+\cdots+\binom{14}{9}\\
&= \left[\binom{7}{1}+\binom{7}{2}\right]+\binom{8}{3}+\cdots+\binom{14}{9}\\
&= \left[\binom{8}{2}+\binom{8}{3}\right]+\binom{9}{4}+\cdots+\binom{14}{9}\\
&= \ldots\\
&= \binom{14}{8}+\binom{14}{9} = \binom{15}{9} = 5\,005,
\end{aligned}$$

ở đây ta áp dụng Định lý 1.2, và lưu ý $\binom{5}{0} = \binom{6}{0} = 1$.

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

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

Ví dụ 1.28. a) Tìm các tổng riêng của $n = 4$.

b) Có bao nhiêu tổng riêng của n?

Giải. a) Các tổng riêng của 4 là

1) 4
2) 3 + 1
3) 1 + 3
4) 2 + 2
5) 2 + 1 + 1
6) 1 + 2 + 1
7) 1 + 1 + 2
8) 1 + 1 + 1 + 1

b) Tổng riêng 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 tổng riêng của n là

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

theo Hệ quả 1.1.

□

Bài tập 1.6

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

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

1.52. 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.

1.53. 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 $x_1, x_2, x_3 > 0$?

1.54. 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?

1.55. 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
```

1.56. 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
```

1.57. 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 $m \geq rn$.

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

1.59. Cho số tự nhiên r và số nguyên dương n. Lập trình liệt kê các nghiệm nguyên không âm của phương trình $x_1 + x_2 + \cdots + x_n = r$.

1.60. * Cho hai số nguyên dương n, r. Lập trình liệt kê tổ hợp lặp chập r của các số $1, 2, \ldots, n$.

1.61. * Lập trình liệt kê các tổng riêng của số nguyên dương n.

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 $1, 2, \ldots, n$.

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

b) Các số 1, 2, 3, 4, 5 có hoán vị 23415 đứng trước hoán vị 23514, tổ hợp {1, 2, 5, 6} đứng trước {1, 3, 4, 5}.

Thuật toán sinh các hoán vị dựa trên thủ tục tìm hoán vị kế tiếp của hoán vị cho trước $a_1 a_2 \dots a_n$.

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

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

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

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

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

Để sinh ra $n!$ hoán vị của $1, 2, \dots, n$, ta xuất phát từ hoán vị nhỏ nhất, là $12 \dots n$, và áp dụng liên tiếp $(n! - 1)$ lần thủ tục trên.

Ví dụ 1.30. Tìm hoán vị liền sau 362541.

Giải.

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

đổi chỗ min, > 2

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

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

□

Thủ tục tìm tổ hợp chập r liền sau tổ hợp $\{a_1, a_2, \dots, a_r\}$ của tập 1, 2,. . . , 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 1.31. Tìm tổ hợp liền sau tổ hợp $\{1, 2, 5, 6\}$ của 1, 2, 3, 4, 5, 6.

Giải. $n = 6, r = 4$

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

□

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 1.32. Xâu liền sau 10001 0<u>**0**</u>111 là 10001 0<u>**1000**</u>.

Bài tập 1.7

1.62. 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.

1.63. 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.

1.64. 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

1.65. Lập trình kiểm tra thứ tự từ điển của hai dãy $a_1, a_2, \dots, a_m$ và $b_1, b_2, \dots, b_n$.

1.66. Lập trình tìm hoán vị liền sau một hoán vị cho trước của 1, 2,. . . , n.

1.67. Từ Bài 1.66, liệt kê tất cả hoán vị đứng sau một hoán vị cho trước.

1.68. Cho các số nguyên $n \geq r \geq 1$. Lập trình tìm tổ hợp chập r liền sau một tổ hợp chập r cho trước của 1, 2,..., n.

1.69. Cho các số nguyên dương $n \geq r \geq 1$. Lập trình tìm các tổ hợp chập r của 1, 2,..., n.

1.70. * Cho số nguyên dương r, và dãy số nguyên dương $n_1, n_2, \ldots, n_r$. Lập trình liệt kê các hoán vị lặp của n_1 số 1, n_2 số 2,..., và n_r số r.

1.71. Lập trình tìm xâu nhị phân liền sau xâu nhị phân cho trước.

1.8 Số Catalan

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	$n^r, \quad n, r \geq 0$
có	không	chỉnh hợp	$P(n,r) = \dfrac{n!}{(n-r)!}, \quad 0 \leq r \leq n$
không	có	tổ hợp lặp	$\dbinom{n+r-1}{r}, \quad n, r \geq 0$
không	không	tổ hợp	$C(n,r) = \dfrac{n!}{r!\,(n-r)!} = \dbinom{n}{r}, \quad 0 \leq r \leq n$

- 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

1.72. 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.

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

1.74. Với mọi số nguyên dương n, chứng minh

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

1.75. 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?

1.76. 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 l := 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 Cách (2). Phần tử nào ràng buộc khó hơn thì chọn trước.

1.2 48, quy tắc cộng

1.3 576, quy tắc nhân

1.4 a) 1024 b) 59 049

1.5 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()
```

1.6

a) $\displaystyle\sum_{k=2}^{n} \frac{1}{k!}$

b) $\displaystyle\sum_{k=1}^{7} (-1)^{k-1} k^3$

c) $\displaystyle\sum_{k=1}^{7} k^2$

d) $\displaystyle\sum_{k=0}^{n} \frac{k+1}{n+k}$

e) $\displaystyle\sum_{k=0}^{n} (-1)^k \frac{n+k}{(2k)!}$

1.7 145

1.8

```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

A = [[ 2,  0, -1],
     [ 1,  3, -2]]
B = [[ 0, -1,  1, 0],
     [ 2,  3, -1, 4],
     [-3,  0, -2, 1]]
matrix_mul(A, B)
```

1.9

```python
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

```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']
```

1.12 a) 40 320 b) 5 040 c) 720

1.13 60

1.14 a) 56

1.15 24

1.16 Xem công thức của $P(n, r)$ ở trang 11.

1.17

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]
```

1.18 Đệ quy $n! = n \cdot (n-1)!$ với $n = 1, 2, \ldots$ trong đó $0! = 1$.

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

factorial(3)   # → 6
```

hoặc

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

hoặc

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

Lặp theo định nghĩa

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

1.19 Xem Ví dụ 1.14.

Cách 1: tìm hoán vị của dãy $a = (a_0, a_1, \ldots, a_{n-1})$. Kiểu dữ liệu trả về của hàm đệ quy `permutations(a)` là dãy các hoán vị, trong đó mỗi hoán vị là một dãy, chẳng hạn `permutations([1])` là `[[1]]`, còn `permutations([1, 2, 3])` là `[[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]]`

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

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

Cách 2: hàm tìm hoán vị của một dãy

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

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

1.20

```
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]]
```

1.21

```
def permutations(a, r):
    if r == 1:
        return [[i] for i in a]
    A = []
    for p in permutations(a, r-1):
        for x in a:
            if x not in p:
                A.append(p + [x])
    return A

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

1.23 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.

1.24 a) 210 b) 792 c) 91 d) 3003

1.25 $(n-1)^2$

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

1.27 $\binom{n}{3}$, $\binom{n}{3} - n - n(n-4)$

1.28

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

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

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

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

1.31 $\displaystyle(m+1)\binom{m+n}{m+1} = (m+1)\frac{(m+n)!}{(m+1)!\,(m+n-m-1)!} = \frac{(m+n)!}{m!\,(n-1)!} = \frac{(m+n)!}{m!\,n!}n = n\binom{m+n}{m}.$

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

1.33 $(1+8)^{50} = x^{100} \Leftrightarrow x = \pm 3$

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

1.35 **Đệ quy**

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

hoặc

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

Quy hoạch động tìm tất hệ số nhị thức bậc n. Giả sử $a^{(n)} = \left(a_0^{(n)}, a_1^{(n)}, \ldots, a_n^{(n)}\right)$ là dãy các hệ số nhị thức $\binom{n}{r}$, $r = \overline{0, n}$. Ta có $a^{(0)} = \left(1\right)$, và với $n \geq 1$

$$a_i^{(n)} = \begin{cases} 1 & \text{nếu } i = 0 \text{ hoặc } n \\ a_i^{(n-1)} + a_{i-1}^{(n-1)} & \text{nếu } i = \overline{n-1 \downarrow 1} \end{cases}$$

```python
def pascal(n):
    if n==0:
        return [1]
    A = pascal(n-1)
    for i in range(n-1, 0, -1):
        A[i] += A[i-1]
    A.append(1)
    return A

pascal(4)    # [1, 4, 6, 4, 1]
```

1.36 **Cách 1:** theo công thức Pascal

```python
def combinations(a, r):
    if r == 1:
        return [[i] for i in a]
    if r == len(a):
```

```python
            return [a]
    C = []
    for c in combinations(a[1:], r-1):
        C.append([a[0]] + c)
    for c in combinations(a[1:], r):
        C.append(c)
    return C

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

Cách 2:

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

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

1.37 Có 12 hoán vị.

1.38

a) 30 240

b) 2520

c) $4! \times 5 \times \binom{5}{3, 2}$

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

1.39

a) 6 720

b) 720

1.40

a) 9 979 200

b) $\binom{10}{2, 2, 2, 3, 1} \times 22 \times 2! = 1\,663\,200$

c) $\binom{6}{2, 1, 2, 1} \times 7 \times \binom{6}{1, 2, 3} = 75\,600$

1.41 $\binom{7}{4, 1, 2} \times \binom{8}{4} = 7\,350$

1.42 $\binom{6}{1, 2, 1, 1, 1} + 2 \times \binom{6}{1, 2, 2, 1} = 720$

1.43 3 432

1.44 Nếu $x = a$, ta phải lên trên $y - b$ bước, và nếu $y = b$, ta cần sang phải $x - a$ bước.

Cách 1: đệ quy theo điểm đến. Để đi đến điểm x, y có hai phương án

1) Đến điểm $(x - 1, y)$ rồi sang phải.

2) Đến điểm $(x, y-1)$ rồi lên trên.

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

walks(2, 1, 7, 4)   # ['UUURRRRR', 'UURURRRR', 'URUURRRR', 'RUUURRRR', 'UURRURRR',...]
```

Cách 2: đệ quy theo điểm xuất phát. Có hai phương án đi từ điểm (a, b)

1) Sang phải đến điểm $(a+1, b)$ rồi đi tiếp đến (x, y).

2) Lên trên đến điểm $(a, b+1)$ rồi đi tiếp đến (x, y).

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

1.45 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 $dx = x-a,\ dy = y-b,\ dz = z-c$.

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

1.48 a) 113 400 b) 718 502 400

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

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

1.50 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 $x_4 \ge 26$

1.51

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

1.52 (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 $n = 82$.

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

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

1.55 $\binom{20+4-1}{4}$

1.56 $10 + \binom{15+3-1}{3}$

1.57 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, $1 \le i \le n$. Mỗi cách xếp như vậy tương ứng với một nghiệm nguyên dương của phương trình $x_1 + x_2 + \cdots + x_n = m$.

1.58 a)

```python
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

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

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

```python
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)
```

1.59 **Cách 1:** Xét các trường hợp $x_1 = i$, với $0 \le i \le r$. Ta được phương trình $x_2 + x_3 + \cdots + x_n = r - i$.

```python
def solutions(n, r):
    if n == 1:
        return [[r]]
    S = []
    for i in range(r+1):  # hoặc range(r, -1, -1) để có nghiệm giảm dần
        for s in solutions(n-1, r-i):
            S.append([i] + s)
    return S

solutions(3, 4)  # [[0, 0, 4], [0, 1, 3], [0, 2, 2], [0, 3, 1], [0, 4, 0], [1, 0, 3], [1, 1, 2], [1, 2, 1], [1, 3, 0], [2, 0, 2],
    [2, 1, 1], [2, 2, 0], [3, 0, 1], [3, 1, 0], [4, 0, 0]]
```

Cách 2: Xét hai trường hợp

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

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

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

1.60 Áp dụng nhận xét ở trang 27 để đưa về Bài 1.59.

1.61 **Cách 1:** biến đổi $x_1 + x_2 + \cdots + x_k = n \Leftrightarrow (x_1 - 1) + (x_2 - 1) + \cdots + (x_k - 1) = n - k$ để đưa về bài toán nghiệm nguyên không âm ở Bài 1.59.

```
def summands(n):
    A = []
    for k in range(1, n+1):
        for a in solutions(k, n-k):
            A.append([i + 1 for i in a])
    return A

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

Cách 2: Nếu tổng riêng chỉ có một số hạng thì số hạng đó là n. Ngược lại, xét số hạng cuối của tổng riêng, bằng k, với $1 \leq k \leq n - 1$. Khi đó, các số hạng ở trước là tổng riêng của $n - k$.

```
def summands(n):
    if n==1:
        return [[1]]
    A = []
    for k in range(1, n):
        for a in summands(n-k):
            a.append(k)
            A.append(a)
    A.append([n])
    return A
```

1.62 14532, 15432, 21345, 23451, 23514, 31452, 31542, 43521, 45213, 45321.

1.63 156423, 165432, 231456, 231465, 234561, 314562, 432561, 435612, 541236, 543216, 654312, 654321

1.64

a) 2134

b) 54132

c) 12534

d) 45312

e) 6714253

f) 31542678

1.65

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

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

1.66

```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]
```

1.67

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

1.68

```python
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]
```

1.69

```
n, r = 6, 4
a = [1 + i for i in range(r)]   # [1, 2, 3, 4]
while a is not None:
    print(a)
    a = next_combinations(n, a)
```

1.71

```
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

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

1.72 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

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

1.74 Khai triển nhị thức $[1 + (-1)]^n$, và lưu ý $\binom{n}{r} = 0$ với $r > n$.

1.75 $\binom{7}{1,1,1,1,3}$. a) 5! b) $\binom{5}{3}4!$

1.76 $10 + 12 \cdot r \cdot 2 + 4\sum_{k=5}^{s}(k-2) + 10 \cdot 6 + 8(t-6) = 24r + 2s^2 - 6s + 8t + 14$

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] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. phiên bản 8. McGraw-Hill Education, 2019. 1118 trang.

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

[7] Watson S. Stewart J. Clegg D. *Calculus: Early Transcendentals*. phiên bản 9. Cengage Learning, 2011. 1421 trang.

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