Một số chủ đề trong Lý thuyết tổ hợp

Lê Xuân Thanh

Mục lục

Mục lục

Định nghĩa

Cho trước:

- Các tập hợp $A_1, \ldots, A_m$
- Sơ đồ sắp xếp S
- Các điều kiện $R_1, \ldots, R_n$

Định nghĩa

Mỗi cách sắp xếp các phần tử trong $A_1, \ldots, A_m$

- tuân theo sơ đồ sắp xếp S, và
- thỏa mãn các điều kiện $R_1, \ldots, R_n$

được gọi là một *cấu hình tổ hợp* trên $A_1, \ldots, A_m$

Ví dụ 1

Cho trước:

- A_1: tập hợp 12 sinh viên nữ
- A_2: tập hợp 20 sinh viên nam
- Sơ đồ sắp xếp S: 4 hàng ngang $\times$ 8 sinh viên
- Điều kiện R_1: vị trí đầu hàng 1 là sinh viên nữ
- Điều kiện R_2: sinh viên nam và nữ đứng xen kẽ nhau ở hàng 2
- Điều kiện R_3: bốn vị trí cuối hàng 3 phải là sinh viên nam

Một số cấu hình tổ hợp hợp lệ:

$$\begin{bmatrix} F & M & M & M & F & F & F & F \\ F & M & F & M & F & M & F & M \\ F & M & M & M & M & M & M & M \\ F & M & M & M & M & M & M & F \end{bmatrix} \qquad \begin{bmatrix} F & M & M & M & F & F & F & F \\ M & F & M & F & M & F & M & F \\ M & M & M & F & M & M & M & M \\ M & M & M & M & M & M & F & F \end{bmatrix}$$

Ví dụ 2: Sudoku

Cho trước:

- $A = \{1, 2, \ldots, 9\}$
- Sơ đồ sắp xếp S:
  lưới ô vuông 9×9, chia thành
  9 khối kích thước 3×3
- Điều kiện R_1:
  mỗi ô chứa một chữ số trong A
- Điều kiện R_2: tất cả chữ số trong A
  xuất hiện trên mỗi hàng
- Điều kiện R_3: tất cả chữ số trong A
  xuất hiện trên mỗi cột
- Điều kiện R_4: tất cả chữ số trong A
  xuất hiện trên mỗi khối

4	7	2	5	3	1	8	6	9
8	5	9	6	4	2	3	1	7
1	6	3	9	8	7	2	5	4
3	1	8	7	2	6	4	9	5
5	9	7	3	1	4	6	8	2
6	2	4	8	5	9	1	7	3
9	3	6	4	7	8	5	2	1
7	4	1	2	6	5	9	3	8
2	8	5	1	9	3	7	4	6

Một bảng Sudoku hợp lệ

Các bài toán cơ bản trong Lý thuyết Tổ hợp

- **Bài toán tồn tại:**
  Tồn tại một cấu hình tổ hợp thỏa mãn các điều kiện cho trước?
- **Bài toán đếm:**
  Có bao nhiêu cấu hình tổ hợp thỏa mãn các điều kiện cho trước?
- **Bài toán liệt kê:**
  Liệt kê mọi cấu hình tổ hợp thỏa mãn các điều kiện cho trước.
- **Tối ưu Tổ hợp:**
  Trong số các cấu hình tổ hợp thỏa mãn các điều kiện cho trước, tìm cấu hình tổ hợp *tốt nhất* theo một nghĩa nào đó

Mục lục

Phát biểu bài toán

Cho trước:

- Các tập hợp $A_1, \ldots, A_m$
- Sơ đồ sắp xếp S
- Các điều kiện $R_1, \ldots, R_n$

Bài toán:

Tồn tại hay không một cấu hình tổ hợp trên $A_1, \ldots, A_m$

- tuân theo sơ đồ sắp xếp S, và
- thỏa mãn các điều kiện $R_1, \ldots, R_n$?

Ví dụ 1: Hình vuông Latin

Phát biểu bài toán:

- Cho trước một lưới ô vuông kích thước $n \times n$ và một tập hợp n ký hiệu khác nhau
- Điền các ký hiệu vào các ô vuông sao cho
  - mỗi ký hiệu xuất hiện chính xác một lần trên mỗi hàng, và
  - mỗi ký hiệu xuất hiện chính xác một lần trên mỗi cột

Lịch sử: Đề xuất bởi Euler[1] (sử dụng các ký hiệu là các chữ cái Latin)

Kết quả: Với mọi n, tồn tại một hình vuông Latin kích thước $n \times n$

A	B	C	D	E
E	A	B	C	D
D	E	A	B	C
C	D	E	A	B
B	C	D	E	A

[1] Leonhard Euler (15.04.1707–18.09.1783): nhà toán học, nhà vật lý, nhà thiên văn, nhà địa lý, nhà logic học, kỹ sư người Thụy Sĩ

Ví dụ 2: Bài toán 36 sĩ quan của Euler

Phát biểu bài toán:

- n trung đoàn, mỗi trung đoàn có n sĩ quan với n cấp bậc khác nhau
- Sắp xếp n^2 sĩ quan vào một lưới ô vuông kích thước $n \times n$ sao cho
  - mỗi hàng có sĩ quan thuộc tất cả các trung đoàn và tất cả các cấp bậc, và
  - mỗi cột có sĩ quan thuộc tất cả các trung đoàn và tất cả các cấp bậc

Phát biểu tương đương:

Hình vuông Euler, hoặc cặp hình vuông Latin trực giao

A	B	C
B	C	A
C	A	B

$+$

1	3	2
2	1	3
3	2	1

$=$

A1	B3	C2
B2	C1	A3
C3	A2	B1

Ví dụ 2: Bài toán 36 sĩ quan của Euler

Phát biểu bài toán:

- n trung đoàn, mỗi trung đoàn có n sĩ quan với n cấp bậc khác nhau
- Sắp xếp n^2 sĩ quan vào một lưới ô vuông kích thước $n \times n$ sao cho
  - mỗi hàng có sĩ quan thuộc tất cả các trung đoàn và tất cả các cấp bậc, và
  - mỗi cột có sĩ quan thuộc tất cả các trung đoàn và tất cả các cấp bậc

Phát biểu tương đương:

Hình vuông Euler, hoặc cặp hình vuông Latin trực giao

A	D	B	C
B	C	A	D
C	B	D	A
D	A	C	B

$+$

2	4	1	3
3	1	4	2
4	2	3	1
1	3	2	4

$=$

A2	D4	B1	C3
B3	C1	A4	D2
C4	B2	D3	A1
D1	A3	C2	B4

Ví dụ 2: Bài toán 36 sĩ quan của Euler

Phát biểu bài toán:

- n trung đoàn, mỗi trung đoàn có n sĩ quan với n cấp bậc khác nhau
- Sắp xếp n^2 sĩ quan vào một lưới ô vuông kích thước $n \times n$ sao cho
  - mỗi hàng có sĩ quan thuộc tất cả các trung đoàn và tất cả các cấp bậc, và
  - mỗi cột có sĩ quan thuộc tất cả các trung đoàn và tất cả các cấp bậc

Phát biểu tương đương:

Hình vuông Euler, hoặc cặp hình vuông Latin trực giao

A	B	C	D	E
C	D	E	A	B
E	A	B	C	D
B	C	D	E	A
D	E	A	B	C

$+$

1	2	3	4	5
4	5	1	2	3
2	3	4	5	1
5	1	2	3	4
3	4	5	1	2

$=$

A1	B2	C3	D4	E5
C4	D5	E1	A2	B3
E2	A3	B4	C5	D1
B5	C1	D2	E3	A4
D3	E4	A5	B1	C2

Ví dụ 2: Bài toán 36 sĩ quan của Euler

Phát biểu bài toán:

- n trung đoàn, mỗi trung đoàn có n sĩ quan với n cấp bậc khác nhau
- Sắp xếp n^2 sĩ quan vào một lưới ô vuông kích thước $n \times n$ sao cho
  - mỗi hàng có sĩ quan thuộc tất cả các trung đoàn và tất cả các cấp bậc, và
  - mỗi cột có sĩ quan thuộc tất cả các trung đoàn và tất cả các cấp bậc

Lịch sử:

- Euler *không giải được* bài toán trong trường hợp $n = 6$
- Euler đặt giả thuyết rằng với $n = 4k + 2$ bài toán vô nghiệm
- Năm 1901, Terry[2] chứng minh rằng với $n = 6$ bài toán vô nghiệm
- Năm 1960, Bose, Shrikhande, và Parker[3] chứng minh rằng bài toán có nghiệm với mọi n trừ trường hợp $n = 2$ và $n = 6$

[2] Gaston Terry (27.09.1843–21.06.1913): nhà toán học người Pháp

[3] R.C. Bose, S.S. Shrikhande, and E.T. Parker. Further results on the construction of mutually orthogonal Latin squares and the falsity of Euler's conjecture, *Canadian Journal of Mathematics*, 12:189–203, 1960

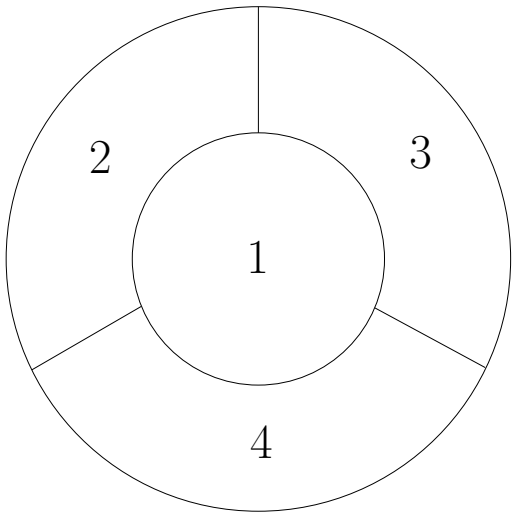
Ví dụ 3: Bài toán 4 màu

Phát biểu bài toán

- Cho trước một bản đồ (phân tách mặt phẳng thành các vùng tiếp giáp nhau), hãy tô màu bản đồ sao cho không có hai vùng kề nhau được tô cùng màu
  - 'Kề nhau' tức là có chung một *đoạn* đường biên giới

Kết quả:

- Mọi bản đồ dều có thể tô bởi $n \geq 5$ màu
- Một số bản đồ không thể tô bởi $n \leq 3$ màu



Ví dụ 3: Bài toán 4 màu

Lịch sử:

- Trong khi cố gắng tô màu bản đồ các hạt của Anh, Guthrie[4] nhận thấy rằng 4 màu là đủ
- Ngày 23.10.1852, trong một lá thư gửi Hamilton[5], De Morgan[6] đặt giả thuyết "*4 màu là đủ tô mọi bản đồ*"
- Năm 1976, Appel[7] và Haken[8] đưa ra một chứng minh sử dụng sự trợ giúp của máy tính

---

[4]Francis Guthrie (22.01.1831–19.10.1899): nhà toán học và thực vật học người Nam Phi

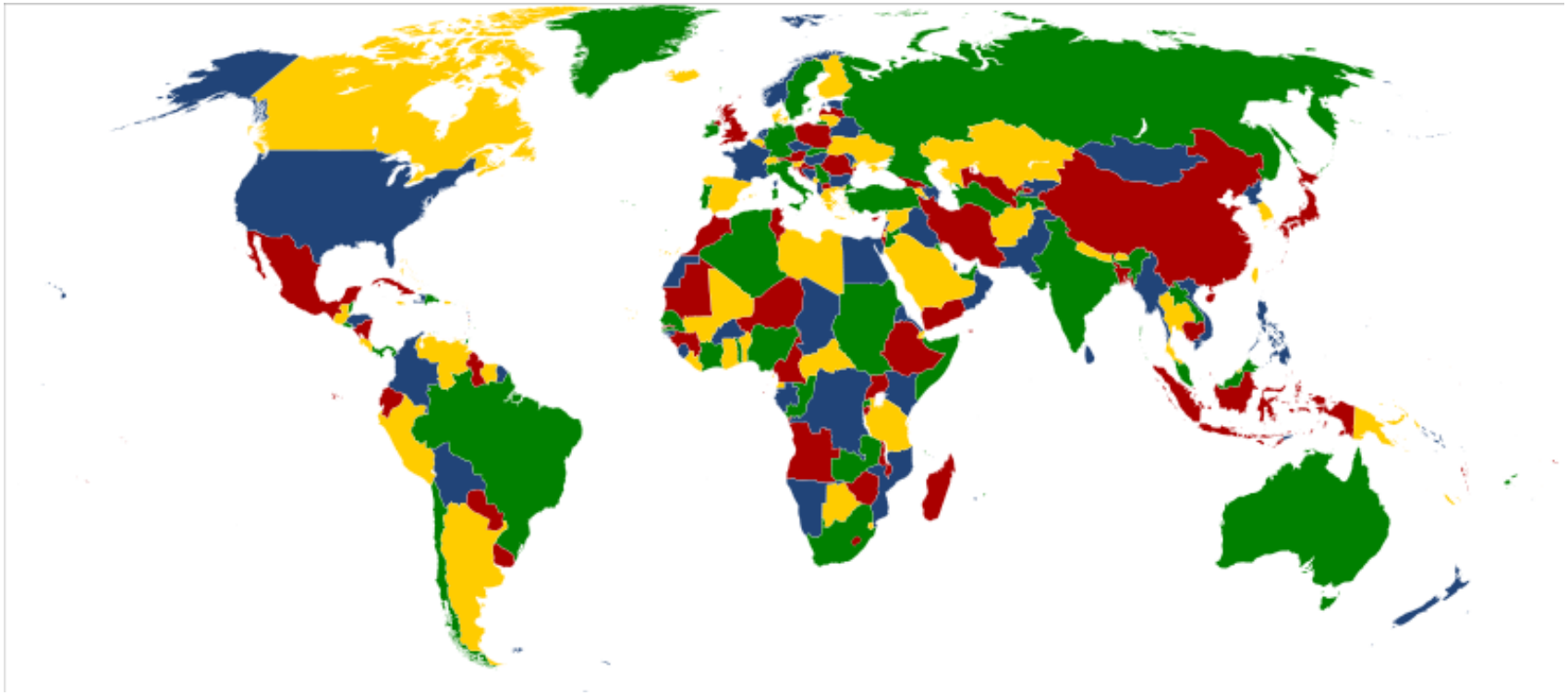
[5]William Rowan Hamilton (03.08.1805–02.09.1865): nhà toán học người Ailen

[6]Augustus De Morgan (27.06.1806–18.03.1871): nhà toán học và logic học người Anh

[7]Kenneth Ira Appel (08.10.1932–19.04.2013): nhà toán học người Mỹ

[8]Wolfgang Haken (21.06.1928): nhà toán học người Đức

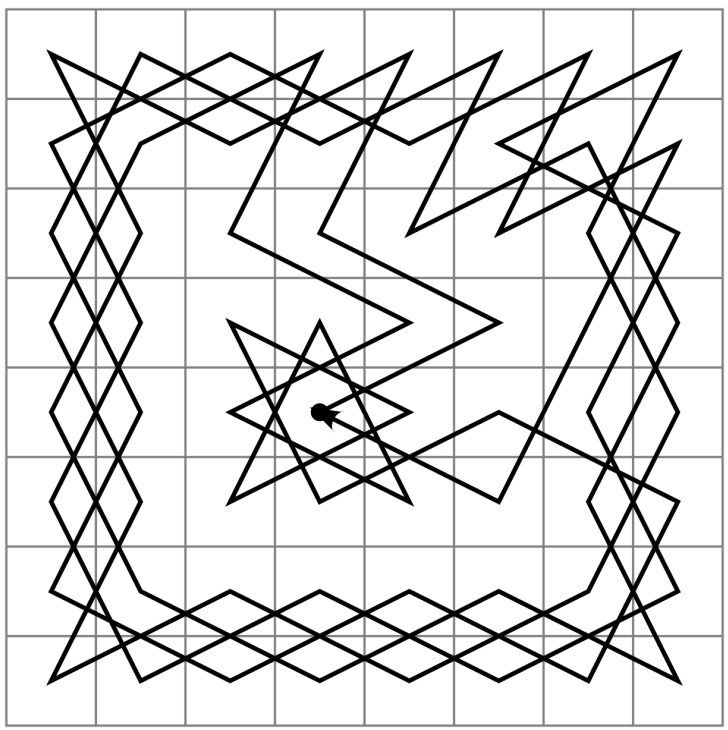
Ví dụ 3: Bài toán 4 màu

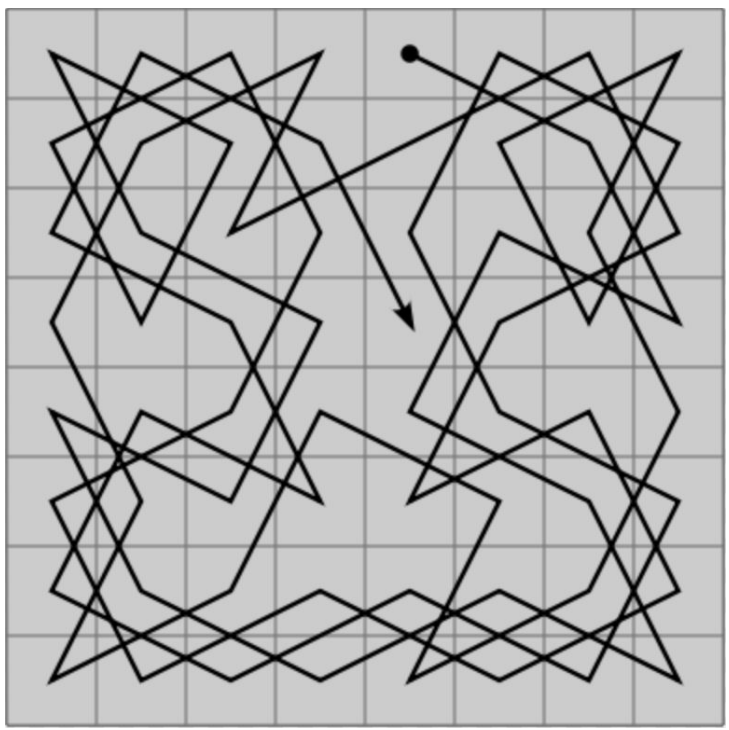


Ví dụ 4: Bài toán quân mã đi tuần

Phát biểu bài toán:

- Cho một bàn cờ kích thước $m \times n$
- Tìm một tua của quân mã đi qua mỗi ô chính xác một lần
  - Tua đóng: từ ô cuối quân mã có thể nhảy một bước về ô đầu
  - Tua mở: không phải tua đóng





Ví dụ 4: Bài toán quân mã đi tuần

Lịch sử:

- Thế kỷ 9: tua của quân mã trên nửa bàn cờ đã được giới thiệu dưới dạng thơ trong tác phẩm *Kavyalankara* (tiếng Phạn) của Rudrata[9]
- Thế kỷ 14: một tua khác của quân mã trên nửa bàn cờ được 'mã hóa' trong hai câu thơ liên tiếp trong tác phẩm *Paduka Sahasram* (tiếng Phạn) của Desika[10]
- 1823: thuật toán đầu tiên tìm tua của quân mã trên bàn cờ được đề xuất bởi Warnsdorff[11]

---

[9] Rudrata (khoảng thế kỷ 9): nhà thơ, nhà lý luận văn học người Kashmir (một khu vực phía Bắc Ấn Độ)

[10] Sri Vedanta Desikan (1268–1369): nhà thơ, nhà triết học, nhà logic, nhà toán học người Ấn Độ

[11] H.C. von Warnsdorff. Des Rösselsprunges einfachste und allgeneinste Lösung, Schmalkalden, Deutschland, 1823.

Ví dụ 4: Bài toán quân mã đi tuần

Một số kết quả:

- Schwenk[12]: với mọi bàn cờ kích thước $m \times n$ với $m \leq n$, luôn có một tua đóng của quân mã, trừ khi
  - hoặc m và n đều lẻ,
  - hoặc $m \in \{1, 2, 4\}$,
  - hoặc $m = 3$ và $n \in 4, 6, 8$
- Conrad et al.[13], Cull and de Curtins[14]: với mọi bàn cờ kích thước $m \times n$ với $m, n \geq 5$, luôn có một tua (có thể là tua mở) của quân mã

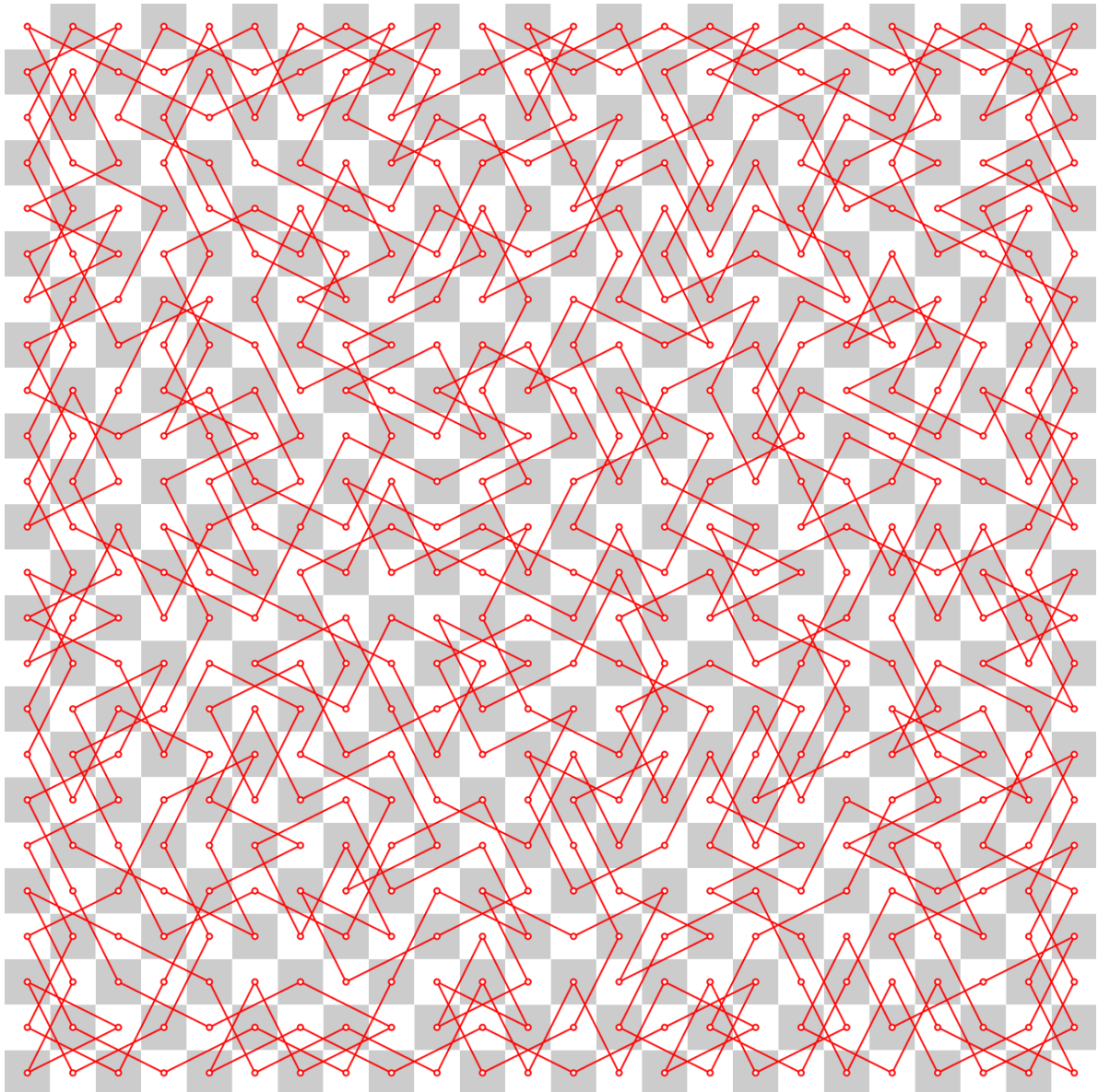
---

[12] A.J. Schwenk. Which rectangular chessboards have a knight's tour? *Mathematics Magazine*, 325–332, 1991

[13] A. Conrad, T. Hindrichs, H. Morsy, and I. Wegener. Solution of the knight's Hamiltonian path problem on chessboards. *Discrete Applied Mathematics*, 50(2):125–134, 1994

[14] P. Cull and J. de Curtins. Knight's tour revisited. *Fibonacci Quarterly*, 16:276–285, 1978

Ví dụ 4: Bài toán quân mã đi tuần



Mục lục

Ý tưởng chính

Bài toán: Chứng minh sự tồn tại của một cấu hình tổ hợp

Ý tưởng chính của phương pháp phản chứng:

- Giả sử điều ngược lại
  (tức là không tồn tại cấu hình tổ hợp như mong muốn)
- Sử dụng các suy luận logic để chỉ ra một mâu thuẫn
  (do đó cấu hình tổ hợp mong muốn phải tồn tại)

Ví dụ 1

Bài toán:

- *Cho trước:* 7 đoạn thẳng có độ dài $\in (10, 100)$
- *Chứng minh:* có thể chọn ra 3 đoạn thẳng tạo thành tam giác

Chứng minh:

- Giả sử điều ngược lại
- Sắp xếp các đoạn thẳng theo thứ tự tăng dần của độ dài, đặt $a_1, \ldots, a_7$ là độ dài các đoạn thẳng
- Theo giả thiết phản chứng:

$$\begin{aligned} a_1 + a_2 \leq a_3 &\quad \Rightarrow \quad a_3 > 20 \\ a_2 + a_3 \leq a_4 &\quad \Rightarrow \quad a_4 > 30 \\ a_3 + a_4 \leq a_5 &\quad \Rightarrow \quad a_5 > 50 \\ a_4 + a_5 \leq a_6 &\quad \Rightarrow \quad a_6 > 80 \\ a_5 + a_6 \leq a_7 &\quad \Rightarrow \quad a_7 > 130 \qquad (*) \end{aligned}$$

- (*) mâu thuẫn với giả thiết $a_7 \in (10, 100)$

Ví dụ 2

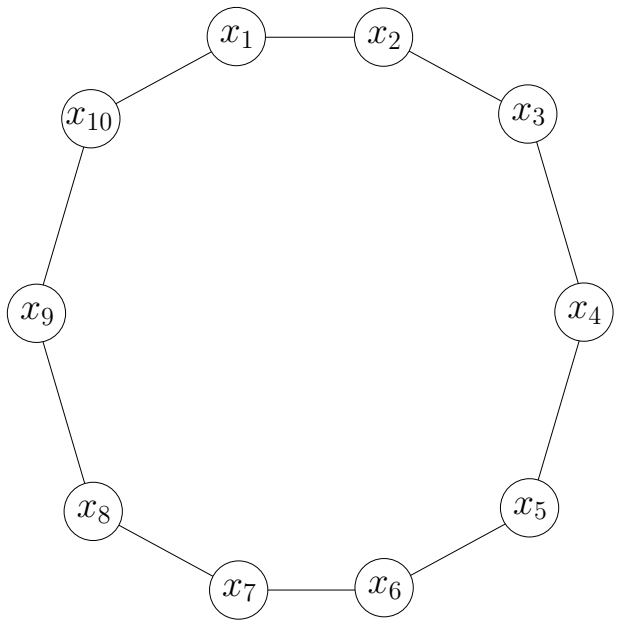
Bài toán:

- *Cho trước:* một thập giác, mỗi đỉnh điền một chữ số
- *Chứng minh:* có 3 đỉnh liên tiếp có tổng > 13

Chứng minh:

- Giả sử điều ngược lại, ta có

$$\begin{array}{l} x_1 + x_2 + x_3 \leq 13 \\ x_2 + x_3 + x_4 \leq 13 \\ \ldots \\ x_9 + x_{10} + x_1 \leq 13 \\ \hline x_{10} + x_1 + x_2 \leq 13 \\ \hline 3(x_1 + \ldots + x_{10}) \leq 130 \end{array} \quad (*)$$



- Tuy nhiên
  $x_1 + \ldots + x_{10} = 0 + 1 + \ldots + 9 = 45$
  mâu thuẫn (*)

Ví dụ 3

Bài toán:

- *Cho trước:* 31 máy tính
- *Chứng minh:* không thể kết nối thành một mạng sao cho mỗi máy tính được kết nối với chính xác 5 máy tính khác

Chứng minh:

- Giả sử điều ngược lại
  (mỗi máy tính có thể kết nối với chính xác 5 máy tính khác)
- Khi đó số kết nối là

$$\frac{31 \times 5}{2} = 75.5 \notin \mathbb{Z}$$

  mâu thuẫn với tính nguyên của số kết nối

Mục lục

Nguyên lý chuồng bồ câu (Pigeonhole principle)

Nguyên lý chuồng bồ câu

Nếu nhốt $k+1$ con chim bồ câu vào k chuồng, thì phải có ít nhất một chuồng chứa nhiều hơn một con chim bồ câu.

Chứng minh.

- Giả sử điều ngược lại (không có chuồng nào chứa nhiều hơn một con chim bồ câu)
- Khi đó tổng số chim bồ câu nhiều nhất là k (mâu thuẫn với giả thiết có $k+1$ con chim bồ câu)

Nguyên lý Dirichlet[15]

Nguyên lý Dirichlet

Nếu n đối tượng được phân hoạch vào k tập hợp, thì phải có ít nhất một tập hợp chứa ít nhất $\left\lceil \frac{n}{k} \right\rceil$ đối tượng.

Chứng minh.

- Giả sử điều ngược lại (trong k tập hợp, không có tập hợp nào chứa nhiều hơn $\left\lceil \frac{n}{k} \right\rceil - 1$ đối tượng)
- Khi đó tổng số đối tượng nhiều nhất là

$$k\left(\left\lceil \frac{n}{k} \right\rceil - 1\right) < k\left(\frac{n}{k} + 1 - 1\right) = n$$

mâu thuẫn với giả thiết có n đối tượng

[15] Johann Peter Gustav Lejeune Dirichlet (13.02.1805–05.05.1859): nhà toán học người Đức

Ví dụ 1

- Có 12 tháng trong một năm, vì vậy trong một lớp có 25 sinh viên luôn có ít nhất $\left\lceil \frac{25}{12} \right\rceil = 3$ sinh viên có sinh nhật trong cùng một tháng
- Có 4 chất trong một bộ bài, do đó trong 13 quân bài luôn có ít nhất $\left\lceil \frac{13}{4} \right\rceil = 4$ quân bài cùng chất
- Trong một ngăn kéo có 5 đôi tất, chọn 6 chiếc tất sẽ đảm bảo có ít nhất 2 chiếc cùng đôi
- Một ánh xạ $f : A \to B$ với $|A| < |B|$ không thể là toàn ánh
- Một ánh xạ $f : A \to B$ với $|A| > |B|$ không thể là đơn ánh

Ví dụ 2

Bài toán: Trong mọi bộ n số nguyên $a_1, \ldots, a_n$, luôn tồn tại các số liên tiếp $a_i, a_{i+1}, \ldots, a_j$ có tổng chia hết cho n.

Chứng minh.

- Xét n tổng của các số liên tiếp trong bộ số đã cho

$$s_1 = a_1, \quad s_2 = a_1 + a_2, \quad \ldots, \quad s_n = a_1 + a_2 + \ldots + a_n.$$

- Nếu một trong các tổng $s_1, \ldots, s_n$ chia hết cho n, ta có điều phải chứng minh.
- Nếu không, xét số dư của các phép chia $s_1, \ldots, s_n$ cho n, ta nhận được n kết quả với $n-1$ giá trị $1, 2, \ldots, n-1$.
  - Tồn tại một cặp s_i, s_j $(i < j)$ có cùng số dư khi chia cho n.
  - Hệ quả là

$$s_j - s_i = a_i + a_{i+1} + \ldots + a_j$$

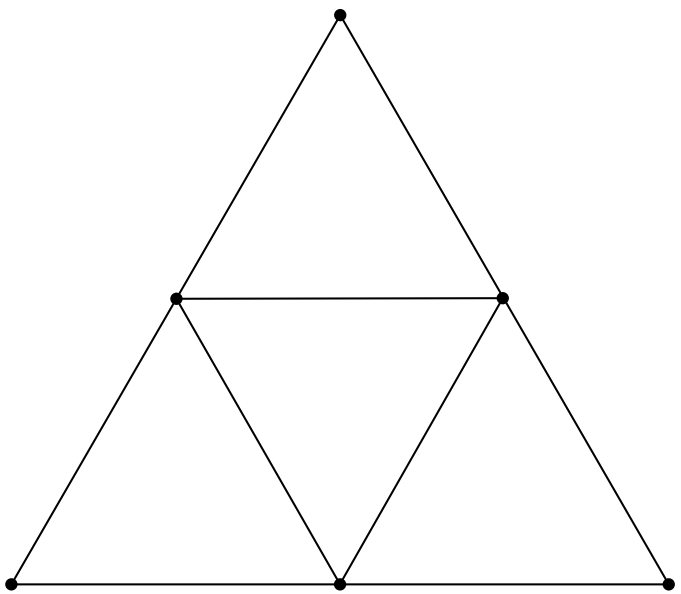
chia hết cho n.

Ví dụ 3

Bài toán: Trong 5 điểm nằm bên trong một tam giác đều có độ dài cạnh bằng 1, luôn có 2 điểm có khoảng cách không quá $\frac{1}{2}$.

Chứng minh.

- Vẽ các đường trung bình của tam giác đã cho. Các đường trung bình này chia tam giác đã cho thành 4 tam giác đều nhỏ hơn (có độ dài cạnh bằng $\frac{1}{2}$).
- Trong số 5 điểm đã cho, có ít nhất 2 điểm nằm trong cùng một tam giác đều nhỏ.
- Chú ý rằng khoảng cách giữa hai điểm trong một tam giác đều nhỏ không vượt quá độ dài cạnh ($\frac{1}{2}$).



Mục lục

Phát biểu bài toán

Cho trước:

- Các tập hợp $A_1, \ldots, A_m$
- Sơ đồ sắp xếp S
- Các điều kiện $R_1, \ldots, R_n$

Định nghĩa

Mỗi cách sắp xếp các phần tử trong $A_1, \ldots, A_m$

- tuân theo sơ đồ sắp xếp S, và
- thỏa mãn các điều kiện $R_1, \ldots, R_n$

được gọi là một *cấu hình tổ hợp* trên $A_1, \ldots, A_m$

Bài toán: Có bao nhiêu cấu hình tổ hợp như vậy?

Một số kết quả không tầm thường

- Số lời giải Sudoku:

$$6\ 670\ 903\ 752\ 021\ 072\ 936\ 960 \quad (\approx 6 \times 10^{21}).$$

- Số tua của quân mã trên bàn cờ kích thước 8×8:

$$19\ 591\ 828\ 170\ 979\ 904 \quad (\approx 19 \times 10^{15}),$$

trong đó số tua đóng là

$$26\ 534\ 728\ 821\ 064 \quad (\approx 26 \times 10^{12}).$$

Các nguyên lý cơ bản

Quy tắc một-một

Nếu có một song ánh giữa hai tập hợp hữu hạn A_1 và A_2, thì $|A_1| = |A_2|$.

Quy tắc tổng

Giả sử một công việc H có thể hoàn thành bằng cách thực hiện *một trong các hành động khác nhau* $H_1, \ldots, H_n$, trong đó hành động H_i có thể được thực hiện theo a_i cách $(i = 1, \ldots, n)$. Khi đó, có $\sum_{i=1}^{n} a_i$ cách hoàn thành công việc H.

Quy tắc nhân

Giả sử một công việc H có thể hoàn thành bằng cách thực hiện *một dãy* n hành động $H_1, \ldots, H_n$, trong đó hành động H_i có thể được thực hiện theo a_i cách $(i = 1, \ldots, n)$. Khi đó, có $\prod_{i=1}^{n} a_i$ cách hoàn thành công việc H.

Các cấu hình cơ bản

- Chỉnh hợp không lặp chập k của n
  (k-permutation of n without repetition)
- Chỉnh hợp có lặp chập k của n
  (k-permutation of n with repetition)
- Tổ hợp không lặp
  (Combination without repetition)
- Tổ hợp có lặp
  (Combination with repetition)
- Hoán vị không lặp
  (Permutation without repetition)
- Hoán vị có lặp
  (Permutation with repetition)

Chỉnh hợp không lặp chập k của n

Cho trước:

- Tập hợp A với $|A| = n$
- Sơ đồ sắp xếp S: bộ có thứ tự k phần tử $(x_1, \ldots, x_k)$
- Điều kiện R: $x_1, \ldots, x_k$ phân biệt

Định nghĩa

Một cấu hình tổ hợp trên A theo sơ đồ sắp xếp S và thỏa mãn điều kiện R được gọi là một *chỉnh hợp không lặp chập k của n phần tử của A*.

Đếm:

- Công việc H: tạo một chỉnh hợp không lặp chập k của n phần tử của A
- H = dãy k hành động $H_1, \ldots, H_k$, trong đó
  H_i = chọn giá trị cho phần tử x_i của bộ $(x_1, \ldots, x_k)$
- Có $|A| = n$ cách chọn x_1,
  $n-1$ cách chọn x_2 (không tính giá trị đã chọn của x_1), ...,
  $n-k+1$ cách chọn x_k (không tính các giá trị đã chọn của $x_1, \ldots, x_{k-1}$).
- Theo quy tắc nhân, số cách hoàn thành công việc H là

$$\begin{cases} \frac{n!}{(n-k)!} & \text{nếu } k \leq n, \\ 0 & \text{nếu } k > n. \end{cases}$$

Chỉnh hợp không lặp chập k của n: Ví dụ

Bài toán:

- Cho trước một danh sách 20 bài tập.
- Chọn 5 bài tập khác nhau từ danh sách này để tạo một tờ bài tập.
- Có thể tạo ra bao nhiêu tờ bài tập khác nhau?

Lời giải:

- Mỗi tờ bài tập là một hoán vị không lặp chập 5 của 20 bài tập trong danh sách
- Số tờ bài tập khác nhau có thể được tạo ra:

$$\frac{20!}{(20-5)!} = \frac{20!}{15!} = 1\ 860\ 480.$$

Các cấu hình cơ bản

- Chỉnh hợp không lặp chập k của n
  (k-permutation of n without repetition)
- Chỉnh hợp có lặp chập k của n
  (k-permutation of n with repetition)
- Tổ hợp không lặp
  (Combination without repetition)
- Tổ hợp có lặp
  (Combination with repetition)
- Hoán vị không lặp
  (Permutation without repetition)
- Hoán vị có lặp
  (Permutation with repetition)

Chỉnh hợp có lặp chập k của n

Cho trước:

- Tập hợp A với $|A| = n$
- Sơ đồ sắp xếp S: bộ có thứ tự k phần tử $(x_1, \ldots, x_k)$
- Điều kiện R: $x_1, \ldots, x_k \in A$

Định nghĩa

Một cấu hình tổ hợp trên A theo sơ đồ sắp xếp S thỏa mãn điều kiện R được gọi là một *chỉnh hợp có lặp chập k của n phần tử của A*.

Đếm:

- Công việc H: tạo một chỉnh hợp có lặp chập k của n phần tử của A
- $H =$ dãy k hành động $H_1, \ldots, H_k$, trong đó
  $H_i =$ chọn giá trị cho phần tử x_i của bộ $(x_1, \ldots, x_k)$
- Mỗi hành động H_i có $|A| = n$ cách thực hiện
- Theo quy tắc nhân, số cách hoàn thành công việc H là

$$P_r(n, k) = |A|^k = n^k$$

Chỉnh hợp có lặp chập k của n: Ví dụ 1

Bài toán: Có bao nhiêu xâu nhị phân độ dài k?

Lời giải:

- Một xâu nhị phân độ dài k có dạng

$$(x_1, x_2, \ldots, x_k) \quad \text{với} \quad x_i \in \{0, 1\}$$

  là một chỉnh hợp có lặp chập k của 2 phần tử $\{0, 1\}$.
- Số xâu nhị phân độ dài k là 2^k.

Chỉnh hợp có lặp chập k của n: Ví dụ 2

Bài toán: Cho trước tập hợp A với $|A| = n$. Có bao nhiêu tập hợp con của A?

Lời giải:

- Gọi $x_1, \ldots, x_n$ là các phần tử của A.
- Mỗi tập hợp con của A tương ứng 1-1 với một xâu nhị phân độ dàin:

$$B \subset A \quad \leftrightarrow \quad b = (b_1, \ldots, b_n) \in \{0,1\}^n$$

trong đó

$$b_i = \begin{cases} 1 & \text{nếu } x_i \in B, \\ 0 & \text{nếu } x_i \notin B. \end{cases}$$

- Có 2^n xâu nhị phân độ dài n,
vì vậy ta có 2^n tập hợp con của A.

Chỉnh hợp có lặp chập k của n: Ví dụ 3

Bài toán:

- Mỗi vé xổ số có mã gồm hai phần:
  - phần đầu gồm 2 chữ cái viết hoa,
  - phần sau bao gồm 5 chữ số.
- Xác suất trúng giải đặc biệt là bao nhiêu?

Solution:

- Có 26 chữ cái viết hoa (bảng chữ cái tiếng Anh), nên có $26^2 = 676$ giá trị cho phần đầu của mỗi vé xổ số.
- Có 10 chữ số, nên có $10^5 = 100\ 000$ giá trị cho phần sau của mỗi vé xổ số.
- Số lượng vé xổ số là

$$676 \times 100\ 000 = 67\ 600\ 000.$$

- Xác suất trúng giải đặc biệt là

$$\frac{1}{67\ 600\ 000} \approx 1.48 \times 10^{-7}.$$

Các cấu hình cơ bản

- Chỉnh hợp không lặp chập k của n
  (k-permutation of n without repetition)
- Chỉnh hợp có lặp chập k của n
  (k-permutation of n with repetition)
- Tổ hợp không lặp
  (Combination without repetition)
- Tổ hợp có lặp
  (Combination with repetition)
- Hoán vị không lặp
  (Permutation without repetition)
- Hoán vị có lặp
  (Permutation with repetition)

Tổ hợp không lặp

Cho trước:

- Tập hợp A với $|A| = n$
- Sơ đồ sắp xếp S: tập hợp k phần tử $\{x_1, \ldots, x_k\}$
- Điều kiện R: $x_1, \ldots, x_k \in A$ phân biệt

Định nghĩa

Một cấu hình tổ hợp trên A theo sơ đồ sắp xếp S thỏa mãn điều kiện R được gọi là một *tổ hợp không lặp chập k của n phần tử của A* (gọi tắt: tổ hợp chập k của n).

Đếm:

- *Ý tưởng:* tổ hợp chập k của n + sắp thứ tự = chỉnh hợp không lặp chập k của n
- Công việc H = "tạo một chỉnh hợp không lặp chập k của n phần tử của A"
  có thể hoàn thành sau khi thực hiện lần lượt các hành động
  - H_1: tạo một tổ hợp chập k của n phần tử của A
  - H_2: tạo một chỉnh hợp không lặp chập k của k phần tử trong H_1
- H có thể hoàn thành theo $P_{wor}(n, k)$ cách,
  H_2 có thể thực hiện theo $P_{wor}(k, k)$ cách,
  nên số cách thực hiện H_1 là

$$C_{wor}(n, k) = \binom{n}{k} := \begin{cases} \frac{P_{wor}(n,k)}{P_{wor}(k,k)} = \frac{n!}{(n-k)!k!} & \text{nếu } 1 \le k \le n, \\ 0 & \text{nếu ngược lại.} \end{cases}$$

Tổ hợp không lặp: Ví dụ

Bài toán:

- Cho trước 10 điểm trên mặt phẳng
  trong đó không có 3 điểm nào nằm trên cùng một đường thẳng.
- Tính số tam giác với các đỉnh thuộc vào 10 điểm đã cho.

Solution:

- Mỗi tam giác tương ứng 1-1 với
  một tổ hợp không lặp chập 3 của 10 điểm đã cho.
- Số tam giác cần tìm là

$$\binom{10}{3} = \frac{10!}{(10-3)!3!} = 120.$$

Các cấu hình cơ bản

- Chỉnh hợp không lặp chập k của n
  (k-permutation of n without repetition)
- Chỉnh hợp có lặp chập k của n
  (k-permutation of n with repetition)
- Tổ hợp không lặp
  (Combination without repetition)
- Tổ hợp có lặp
  (Combination with repetition)
- Hoán vị không lặp
  (Permutation without repetition)
- Hoán vị có lặp
  (Permutation with repetition)

Tổ hợp có lặp

Cho trước:

- Tập hợp A với $|A| = n$
- Sơ đồ sắp xếp S: đa tập k phần tử $\{x_1, \ldots, x_k\}$
- Điều kiện R: $x_1, \ldots, x_k \in A$

Định nghĩa

Một cấu hình tổ hợp trên A theo sơ đồ sắp xếp S thỏa mãn điều kiện R được gọi là một *tổ hợp có lặp chập k của n phần tử của A*.

Đếm:

- Gọi $a_1, \ldots, a_n$ là các phần tử của A
- Mỗi tổ hợp có lặp chập k của n phần tử của A tương ứng 1-1 với một bộ có thứ tự $(f_1, \ldots, f_n) \in \mathbb{Z}_+^n$ với $f_1 + \ldots + f_n = k$, trong đó f_i là số lần xuất hiện của a_i trong tổ hợp
- Mỗi bộ có thứ tự $(f_1, \ldots, f_n) \in \mathbb{Z}_+^n$ với $f_1 + \ldots + f_n = k$ tương ứng 1-1 với một xâu nhị phân

$$\underbrace{0\ldots0}_{f_1 \text{ lần}}1\underbrace{0\ldots0}_{f_2 \text{ lần}}10\ldots01\underbrace{0\ldots0}_{f_n \text{ lần}}$$

- Số xâu nhị phân như vậy
  $=$ số cách chọn k vị trí cho chữ số 0 trong tổng cộng $n+k-1$ vị trí
  $= \binom{n+k-1}{k} = \frac{(n+k-1)!}{(n-1)!k!} =: C_r(n,k)$

Tổ hợp có lặp: Ví dụ 1

Bài toán: Có 4 loại bút, cần mua 20 bút.
Có bao nhiêu phương án mua?

Lời giải:

- Mỗi phương án mua 20 bút với 4 loại bút là một tổ hợp có lặp chập 20 của 4 loại bút.
- Số phương án mua là

$$C_r(4, 20) = \frac{(4 + 20 - 1)!}{20!(4 - 1)!} = \frac{23!}{20!3!} = 1\ 771.$$

Tổ hợp có lặp: Ví dụ 2

Bài toán: Phương trình $x_1 + x_2 + x_3 = 10$
có bao nhiêu nghiệm nguyên không âm?

Lời giải:

- Mỗi nghiệm $(x_1, x_2, x_3) \in \mathbb{Z}_+^3$ của $x_1 + x_2 + x_3 = 10$ tương ứng với một tổ hợp chập 10 của 3.
- Số nghiệm nguyên không âm của phương trình đã cho là

$$C_r(3, 10) = \frac{(3 + 10 - 1)!}{10!(3 - 1)!} = \frac{12!}{10!2!} = 66.$$

Các cấu hình cơ bản

- Chỉnh hợp không lặp chập k của n
  (k-permutation of n without repetition)
- Chỉnh hợp có lặp chập k của n
  (k-permutation of n with repetition)
- Tổ hợp không lặp
  (Combination without repetition)
- Tổ hợp có lặp
  (Combination with repetition)
- Hoán vị không lặp
  (Permutation without repetition)
- Hoán vị có lặp
  (Permutation with repetition)

Hoán vị không lặp

Cho trước:

- Tập hợp A với $|A| = n$
- Sơ đồ sắp xếp S: bộ có thứ tự n phần tử $(x_1, \ldots, x_n)$
- Điều kiện R: $x_1, \ldots, x_n$ phân biệt

Định nghĩa

Một cấu hình tổ hợp trên A theo sơ đồ sắp xếp S thỏa mãn điều kiện R được gọi là một *hoán vị không lặp của n phần tử của A*.

Đếm:

- Một hoán vị không lặp của n phần tử của A
  là một chỉnh hợp không lặp chập n của n phần tử của A.
- Số hoán vị không lặp của n phần tử của A là

$$P_n = P_{wor}(n, n) = n!$$

Hoán vị không lặp: Ví dụ

Bài toán: Có bao nhiêu cách sắp xếp 5 người trên một hàng ngang?

Lời giải:

- Mỗi cách sắp xếp 5 người trên một hàng ngang là một hoán vị không lặp của 5 người đó.
- Số cách sắp xếp là

$$P_5 = 5! = 120.$$

Các cấu hình cơ bản

- Chỉnh hợp không lặp chập k của n
  (k-permutation of n without repetition)
- Chỉnh hợp có lặp chập k của n
  (k-permutation of n with repetition)
- Tổ hợp không lặp
  (Combination without repetition)
- Tổ hợp có lặp
  (Combination with repetition)
- Hoán vị không lặp
  (Permutation without repetition)
- Hoán vị có lặp
  (Permutation with repetition)

Hoán vị có lặp

Cho trước:

- Tập hợp $A = \{a_1, \ldots, a_n\}$
- Bộ có thứ tự $(m_1, \ldots, m_n) \in \mathbb{Z}_+^n$ với $m_1 + \ldots + m_n = m > 0$
- Sơ đồ sắp xếp S: bộ có thứ tự m phần tử $\mathbf{x} = (x_1, \ldots, x_m)$
- Điều kiện R_1: $x_1, \ldots, x_m \in A$
- Điều kiện R_2: a_i xuất hiện chính xác m_i lần $(i = 1, \ldots, n)$

Định nghĩa

Một cấu hình tổ hợp trên A theo sơ đồ sắp xếp S thỏa mãn điều kiện R_1 và R_2 được gọi là một *hoán vị có lặp chập* $(m_1, \ldots, m_n)$ *của* n *phần tử của* A.

Chú ý: hoán vị không lặp tương ứng với $m_1 = \ldots = m_n = 1$.

Đếm:

- Công việc $H =$ "tạo một hoán vị có lặp chập $(m_1, \ldots, m_n)$ của n phần tử của A" có thể hoàn thành sau khi thực hiện liên tiếp các hành động:
  - H_1: gán a_1 vào m_1 thành phần trong $\mathbf{x}$ $\rightsquigarrow$ $\binom{m}{m_1}$ cách
  - H_2: gán a_2 vào m_2 thành phần *khác* trong $\mathbf{x}$ $\rightsquigarrow$ $\binom{m-m_1}{m_2}$ cách
  - $\ldots$
  - H_n: gán a_n vào m_n thành phần *còn lại* trong $\mathbf{x}$ $\rightsquigarrow$ $\binom{m-m_1-\ldots-m_{n-1}}{m_n}$ cách
- Số cách hoàn thành H là

$$P_n(m_1, \ldots, m_n) = \binom{m}{m_1}\binom{m-m_1}{m_2}\ldots\binom{m-m_1-\ldots-m_{n-1}}{m_n} = \frac{m!}{m_1!m_2!\ldots m_n!}.$$

Hoán vị có lặp: Ví dụ 1

Bài toán: Có bao nhiêu số tự nhiên được tạo thành bởi hai chữ số 1, hai chữ số 2, một chữ số 3?

Lời giải:

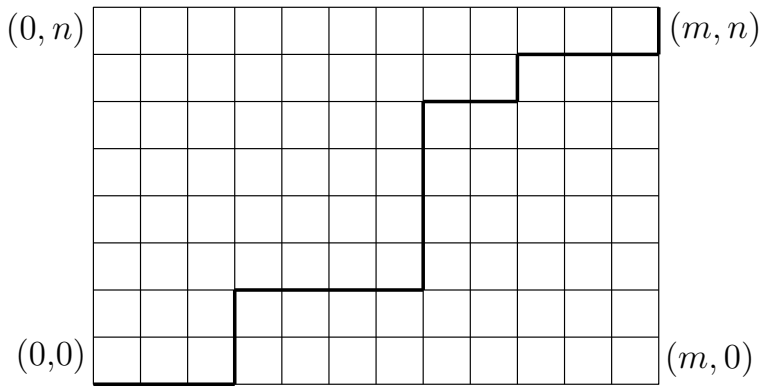
- Mỗi số như vậy tương ứng 1-1 với một hoán vị có lặp chập $(2, 2, 1)$ của các phần tử $\{1, 2, 3\}$.
- Đáp số là

$$P_3(2, 2, 1) = \frac{5!}{2!2!1!} = 30.$$

Hoán vị có lặp: Ví dụ 2

Bài toán:

- Cho một lưới ô vuông kích thước $m \times n$
- Đường tăng dần:
  - Xuất phát tại $(0, 0)$, kết thúc tại (m, n)
  - Mỗi bước: hoặc lên trên, hoặc sang phải tới mắt lưới gần nhất
- Có bao nhiêu đường như vậy?



Lời giải:

- Mỗi đường tăng dần gồm $m + n$ đoạn, tương ứng 1-1 với một xâu nhị phân độ dài $m + n$:

$$\mathbf{b} = (b_1, b_2, \ldots, b_{m+n}) \in \{0, 1\}^{m+n}$$

trong đó

$$b_i = \begin{cases} 0 & \text{nếu sang phải tại bước } i \\ 1 & \text{nếu lên trên tại bước } i \end{cases}$$

với chính xác m thành phần bằng 0 và n thành phần bằng 1.

- Mỗi xâu nhị phân $\mathbf{b}$ như vậy là một hoán vị có lặp chập (m, n) của 2 phần tử $\{0, 1\}$, vì vậy số đường tăng dần là

$$\frac{(m+n)!}{m!n!}.$$

Mục lục

Nguyên lý bù trừ (Inclusion-exclusion principle)

Cho trước:

- Cho $A_1, \ldots, A_n$ là các tập hợp hữu hạn, $I = \{1, \ldots, n\}$.
- Với $i = 1, \ldots, n$, đặt

$$\mathcal{B}_i = \left\{ B \mid B = \bigcap_{k \in J, |J| = i} A_k \right\}$$

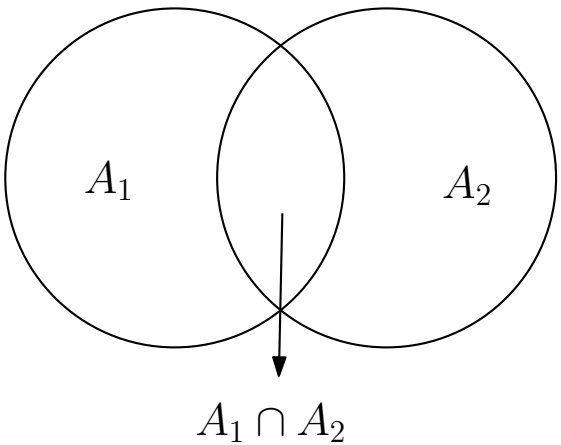
và đặt

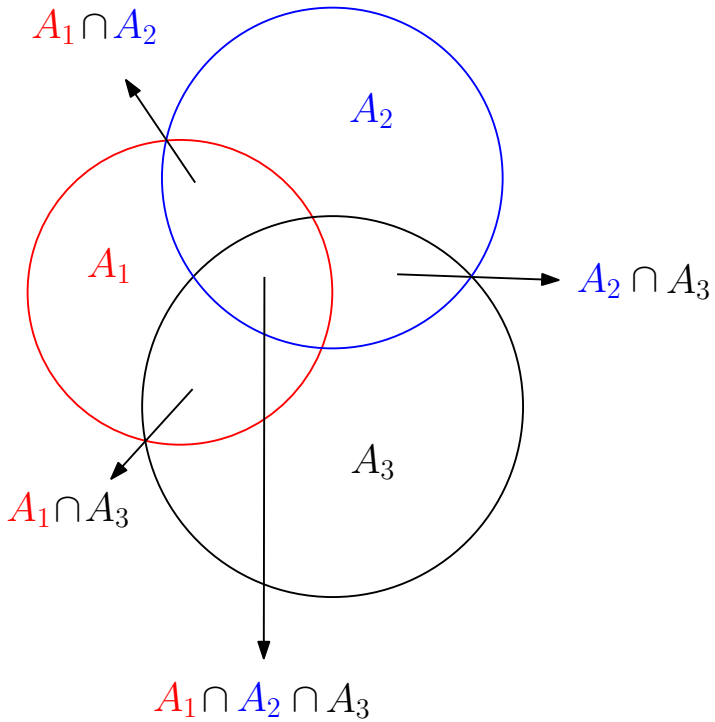
$$c_i = \sum_{B \in \mathcal{B}_i} |B|.$$

Nguyên lý bù trừ

$$|A_1 \cup A_2 \cup \ldots \cup A_n| = \sum_{j=1}^{n} (-1)^{j-1} c_j.$$

Nguyên lý bù trừ: trường hợp riêng





$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

$$\begin{aligned} &|A_1 \cup A_2 \cup A_3| \\ &= |A_1| + |A_2| + |A_3| \\ &-|A_1 \cap A_2| - |A_1 \cap A_3| - |A_2 \cap A_3| \\ &+|A_1 \cap A_2 \cap A_3| \end{aligned}$$

Nguyên lý bù trừ: Ví dụ 1

Bài toán: Có bao nhiêu số tự nhiên không vượt quá 1000 và chia hết cho 7 hoặc 11?

Lời giải:

- Đặt $A :=$ tập hợp các số tự nhiên không vượt quá 1000 và chia hết cho 7, ta có

$$|A| = \left\lfloor \frac{1000}{7} \right\rfloor = 142.$$

- Đặt $B :=$ tập hợp các số tự nhiên không vượt quá 1000 và chia hết cho 11, ta có

$$|B| = \left\lfloor \frac{1000}{11} \right\rfloor = 90.$$

- $A \cap B =$ tập hợp các số tự nhiên không vượt quá 1000 và chia hết cho cả 7 và 11, ta có

$$|A \cap B| = \left\lfloor \frac{1000}{77} \right\rfloor = 12.$$

- $A \cup B =$ tập hợp các số tự nhiên không vượt quá 1000 và chia hết cho 7 hoặc 11. Theo nguyên lý bù trừ, ta có

$$|A \cup B| = |A| + |B| - |A \cap B| = 142 + 90 - 12 = 220.$$

Nguyên lý bù trừ: Ví dụ 2

Bài toán:

- 2092 sinh viên đăng ký học các môn Toán học kỳ này.
- 1232 sinh viên đăng ký học Toán rời rạc.
- 879 sinh viên đăng ký học Giải tích.
- 114 sinh viên đăng ký học Tối ưu.
- 103 sinh viên đăng ký học cả Toán rời rạc và Giải tích.
- 23 sinh viên đăng ký học cả Giải tích và Tối ưu
- 14 sinh viên đăng ký học cả Tối ưu và Toán rời rạc.
- Có bao nhiêu sinh viên đăng ký học cả 3 môn?

Lời giải:

- Gọi D là tập hợp các sinh viên đăng ký học Toán rời rạc,
  gọi A là tập hợp các sinh viên đăng ký học Giải tích,
  gọi O là tập hợp các sinh viên đăng ký học Tối ưu.
- Ta có $|D \cup A \cup O| = 2092$ và

$$|D| = 1232, \quad |A| = 879, \quad |O| = 114, \quad |D \cap A| = 103, \quad |A \cap O| = 23, \quad |D \cap O| = 14.$$

- Số sinh viên đăng ký học cả 3 môn là

$$\begin{aligned} |D \cap A \cap O| &= |D \cup A \cup O| - |D| - |A| - |O| + |D \cap A| + |A \cap O| + |D \cap O| \\ &= 2092 - 1232 - 879 - 114 + 103 + 23 + 14 \\ &= 7. \end{aligned}$$

Mục lục

Công thức truy hồi

Định nghĩa

Một công thức truy hồi là một phương trình biểu diễn mỗi phần tử trong một dãy số như một hàm của các phần tử trước nó.

- Công thức truy hồi bậc nhất trên $\{x_n\}_{n\in\mathbb{N}} \subset X$:

$$x_n = \varphi(n, x_{n-1}) \quad \text{với } \varphi : \mathbb{N} \times X \to X$$

  - *Ví dụ:* hàm giai thừa được biểu diễn bởi công thức truy hồi

$$n! = n(n-1)! \quad \text{với } n > 0$$

  và điều kiện ban đầu $0! = 1$.

Công thức truy hồi

Định nghĩa

Một công thức truy hồi là một phương trình biểu diễn mỗi phần tử trong một dãy số như một hàm của các phần tử trước nó.

- Công thức truy hồi bậc k trên $\{x_n\}_{n\in\mathbb{N}} \subset X$:

$$x_n = \varphi(n, x_{n-1}, x_{n-2}, \ldots, x_{n-k}) \quad \text{với } \varphi : \mathbb{N} \times X^k \to X$$

  - *Ví dụ 1:* Dãy Fibonacci được định nghĩa bởi công thức truy hồi

$$F_n = F_{n-1} + F_{n-2}$$

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

Công thức truy hồi

Định nghĩa

Một công thức truy hồi là một phương trình biểu diễn mỗi phần tử trong một dãy số như một hàm của các phần tử trước nó.

- Công thức truy hồi bậc k trên $\{x_n\}_{n\in\mathbb{N}} \subset X$:

$$x_n = \varphi(n, x_{n-1}, x_{n-2}, \ldots, x_{n-k}) \quad \text{với } \varphi : \mathbb{N} \times X^k \to X$$

- *Ví dụ 2:* Hệ số nhị thức được định nghĩa bởi công thức truy hồi

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

và điều kiện ban đầu $\binom{n}{0} = \binom{n}{n} = 1$.

Đếm bằng phương pháp truy hồi

Sử dụng các công thức truy hồi để giải bài toán đếm

- Thiết lập công thức truy hồi
  - *Lược đồ:* xác định các tham số nào là chỉ số, xây dựng các giá trị đếm dưới dạng công thức truy hồi
- Giải công thức truy hồi tuyến tính
  - Thuần nhất với hệ số hằng
  - Không thuần nhất với hệ số hằng

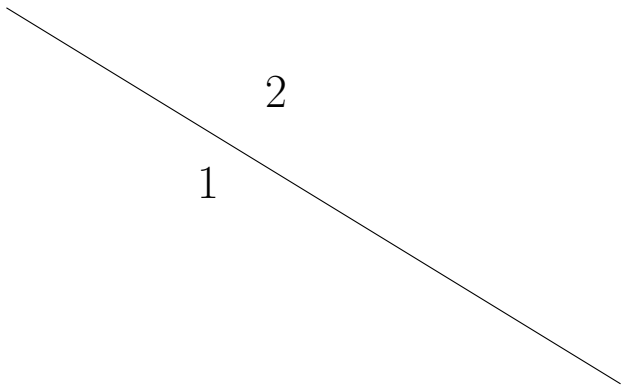
Thiết lập công thức truy hồi: Ví dụ 1

Bài toán:

- Cho trước n đường thẳng trên mặt phẳng thỏa mãn
  - không có cặp đường thẳng nào song song, và
  - không có ba đường thẳng nào đồng quy.
- Các đường thẳng này chia mặt phẳng thành bao nhiêu miền?

Thiết lập công thức truy hồi:

- Gọi S_n là số miền phân hoạch bởi n đường thẳng, ta có $S_1 = 2$.
- Đường thẳng $n+1$ giao với n đường thẳng trước đó tại n giao điểm, làm tăng thêm $n+1$ miền.
- Vì vậy $S_{n+1} = S_n + (n+1)$ với $S_1 = 2$.



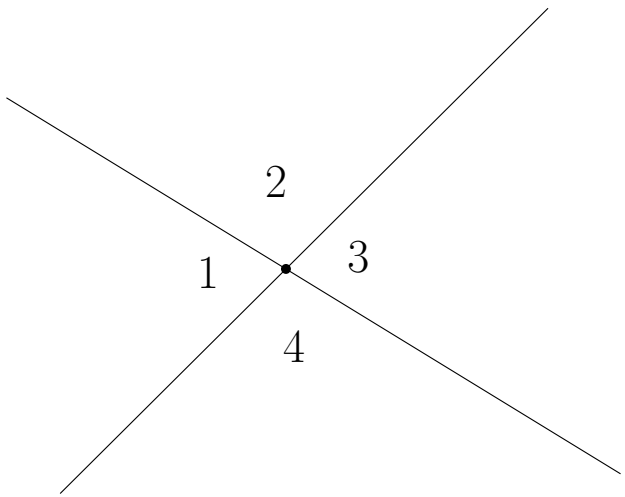
Thiết lập công thức truy hồi: Ví dụ 1

Bài toán:

- Cho trước n đường thẳng trên mặt phẳng thỏa mãn
  - không có cặp đường thẳng nào song song, và
  - không có ba đường thẳng nào đồng quy.
- Các đường thẳng này chia mặt phẳng thành bao nhiêu miền?

Thiết lập công thức truy hồi:

- Gọi S_n là số miền phân hoạch bởi n đường thẳng, ta có $S_1 = 2$.
- Đường thẳng $n+1$ giao với n đường thẳng trước đó tại n giao điểm, làm tăng thêm $n+1$ miền.
- Vì vậy $S_{n+1} = S_n + (n+1)$ với $S_1 = 2$.



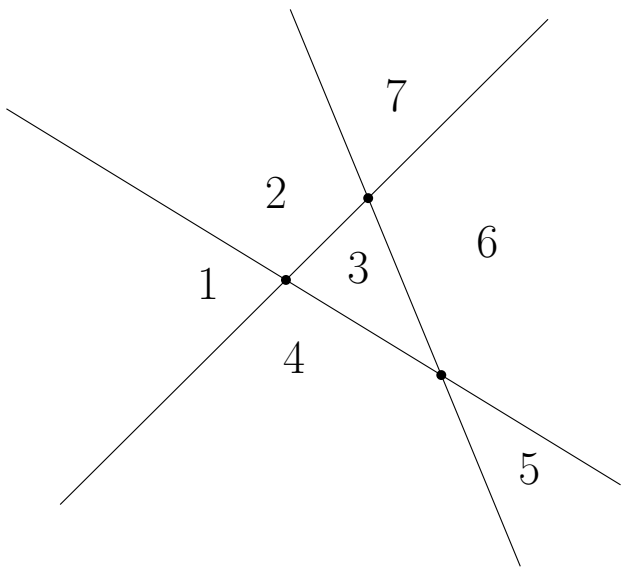
Thiết lập công thức truy hồi: Ví dụ 1

Bài toán:

- Cho trước n đường thẳng trên mặt phẳng thỏa mãn
  - không có cặp đường thẳng nào song song, và
  - không có ba đường thẳng nào đồng quy.
- Các đường thẳng này chia mặt phẳng thành bao nhiêu miền?

Thiết lập công thức truy hồi:

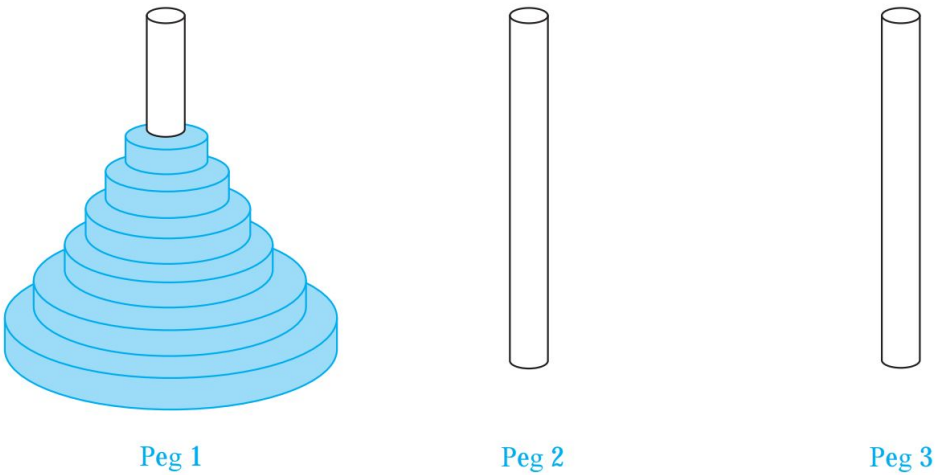
- Gọi S_n là số miền phân hoạch bởi n đường thẳng, ta có $S_1 = 2$.
- Đường thẳng $n+1$ giao với n đường thẳng trước đó tại n giao điểm, làm tăng thêm $n+1$ miền.
- Vì vậy $S_{n+1} = S_n + (n+1)$ với $S_1 = 2$.



Thiết lập công thức truy hồi: Ví dụ 2

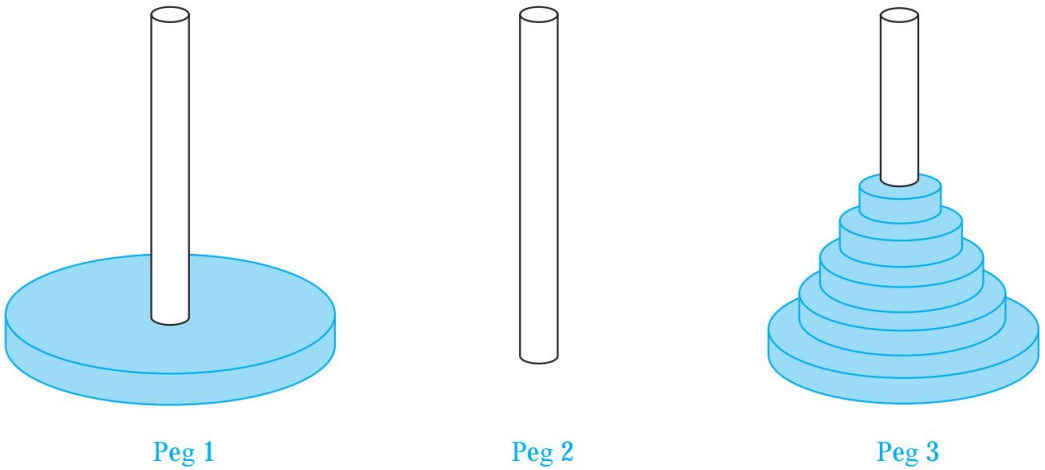
Bài toán tháp Hà Nội[16]:

- Cho trước 3 cọc và n đĩa có kích cỡ khác nhau
- Đặt các đĩa vào cọc 1 theo thứ tự nhỏ dần, đĩa to nhất ở dưới cùng
- Tại mỗi bước chỉ được chuyển một đĩa sang cọc khác
- Không đặt lên trên đĩa nhỏ hơn
- Tìm số bước ít nhất để chuyển tất cả đĩa sang cọc 2



[16]Phát minh năm 1883 bởi Édouard Lucas (04.04.1842–03.10.1891) - nhà toán học người Pháp

Thiết lập công thức truy hồi: Ví dụ 2



Thiết lập công thức truy hồi:

- Gọi S_n là số bước chuyển ít nhất với n đĩa, ta có $S_1 = 1$.
- Công việc "chuyển tất cả đĩa từ cọc 1 sang cọc 2"
  có thể hoàn thành qua các giai đoạn
  - chuyển $n-1$ đĩa nhỏ nhất sang cọc 3 $\rightsquigarrow$ S_{n-1} bước,
  - chuyển đĩa lớn nhất từ cọc 1 sang cọc 2 $\rightsquigarrow$ 1 bước,
  - chuyển $n-1$ đĩa nhỏ nhất từ cọc 3 sang cọc 2 $\rightsquigarrow$ S_{n-1} bước.
- Ta có $S_n = 2S_{n-1} + 1$ với $S_1 = 1$.

Đếm bằng phương pháp truy hồi

Sử dụng các công thức truy hồi để giải bài toán đếm

- Thiết lập công thức truy hồi
  - *Lược đồ:* xác định các tham số nào là chỉ số, xây dựng các giá trị đếm dưới dạng công thức truy hồi
- Giải công thức truy hồi tuyến tính
  - Thuần nhất với hệ số hằng
  - Không thuần nhất với hệ số hằng

Công thức truy hồi tuyến tính thuần nhất với hệ số hằng

Định nghĩa

Một *công thức truy hồi tuyến tính thuần nhất bậc k với hệ số hằng* có dạng

$$x_n = c_1 x_{n-1} + c_2 x_{n-2} + \ldots + c_k x_{n-k},$$

trong đó $c_1, \ldots, c_k$ là các hệ số thực, và $c_k \neq 0$.

Ghi chú:

- **Tuyến tính:** vế phải là một hàm tuyến tính theo k thành phần
- **Thuần nhất:** mỗi số hạng có dạng đơn thức bậc nhất
- **Hệ số hằng:** $c_1, \ldots, c_k$ là các hằng số thực

Ví dụ:

- Bậc 1: $x_n = 2x_{n-1}$
- Bậc 2: $x_n = x_{n-1} + x_{n-2}$

Công thức truy hồi tuyến tính thuần nhất với hệ số hằng

Trường hợp bậc 2:

$$\begin{cases} x_n = c_1 x_{n-1} + c_2 x_{n-2} & (1) \\ x_0 = a_0, x_1 = a_1 & (2) \end{cases}$$

Phương trình đặc trưng: $r^2 - c_1 r - c_2 = 0$ (3)

Định lý

- Nếu (3) có hai nghiệm thực phân biệt r_1, r_2, thì $\{x_n\}$ là nghiệm của (1) khi và chỉ khi

$$x_n = \alpha_1 r_1^n + \alpha_2 r_2^n \quad \text{với } n \in \mathbb{N}, \alpha_1, \alpha_2 \in \mathbb{R}.$$

- Nếu (3) có nghiệm thực duy nhất r_0, thì $\{x_n\}$ là một nghiệm của (1) khi và chỉ khi

$$x_n = \alpha_1 r_0^n + \alpha_2 n r_0^n \quad \text{với } n \in \mathbb{N}, \alpha_1, \alpha_2 \in \mathbb{R}.$$

Chú ý: α_1, α_2 được xác định bởi a_0, a_1.

Công thức truy hồi tuyến tính thuần nhất với hệ số hằng

Ví dụ 1: Dãy Fibonacci

$$\begin{cases} x_n = x_{n-1} + x_{n-2} \\ x_0 = 0, x_1 = 1 \end{cases}$$

Phương trình đặc trưng $r^2 - r - 1 = 0$ có hai nghiệm thực phân biệt

$$r_1 = \frac{1+\sqrt{5}}{2}, \quad r_2 = \frac{1-\sqrt{5}}{2}.$$

Vậy $x_n = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$. Vì $x_0 = 0, x_1 = 1$, ta có

$$\begin{cases} \alpha_1 + \alpha_2 = 0 \\ \frac{1+\sqrt{5}}{2}\alpha_1 + \frac{1-\sqrt{5}}{2}\alpha_2 = 1 \end{cases} \Leftrightarrow \begin{cases} \alpha_1 = \frac{1}{\sqrt{5}} \\ \alpha_2 = -\frac{1}{\sqrt{5}} \end{cases}$$

Công thức hiển:

$$x_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Công thức truy hồi tuyến tính thuần nhất với hệ số hằng

Ví dụ 2: Tìm công thức hiển cho nghiệm của

$$\begin{cases} x_n = 6x_{n-1} - 9x_{n-2} \\ x_0 = 1, x_1 = 6 \end{cases}$$

Phương trình đặc trưng $r^2 - 6r + 9 = 0$ có nghiệm thực duy nhất $r_0 = 3$. Vậy $x_n = \alpha_1 3^n + \alpha_2 n3^n$. Vì $x_0 = 1, x_1 = 6$, ta có

$$\begin{cases} \alpha_1 = 1 \\ 3\alpha_1 + 3\alpha_2 = 6 \end{cases} \Leftrightarrow \begin{cases} \alpha_1 = 1 \\ \alpha_2 = 1 \end{cases}$$

Công thức hiển:

$$x_n = 3^n + n3^n = (n+1)3^n.$$

Công thức truy hồi tuyến tính thuần nhất với hệ số hằng

Trường hợp bậc k:

$$\begin{cases} x_n = c_1 x_{n-1} + c_2 x_{n-2} + \ldots + c_k x_{n-k} & (1) \\ x_0 = a_0, x_1 = a_1, \ldots, x_{k-1} = a_{k-1} & (2) \end{cases}$$

Phương trình đặc trưng: $r^k - c_1 r^{k-1} - c_2 r^{k-2} - \ldots - c_k = 0 \quad (3)$

Định lý

- Nếu (3) có k nghiệm thực phân biệt $r_1, r_2, \ldots, r_k$, thì $\{x_n\}$ là nghiệm của (1) khi và chỉ khi

$$x_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \ldots + \alpha_k r_k^n \quad \text{với } n \in \mathbb{N}, \alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{R}.$$

- Nếu (3) có t nghiệm thực phân biệt r_i với bội m_i $(i = 1, \ldots, t)$ thỏa mãn $m_i \geq 1$ và $m_1 + \ldots + m_t = k$, thì

$$\begin{aligned} x_n = {} & (\alpha_{1,0} + \alpha_{1,1} n + \ldots + \alpha_{1,m_1-1} n^{m_1-1}) r_1^n \\ & + (\alpha_{2,0} + \alpha_{2,1} n + \ldots + \alpha_{2,m_2-1} n^{m_2-1}) r_2^n \\ & + \ldots + (\alpha_{t,0} + \alpha_{t,1} n + \ldots + \alpha_{t,m_t-1} n^{m_t-1}) r_t^n \quad \text{với } n \in \mathbb{N}, \alpha_{ij} \in \mathbb{R}. \end{aligned}$$

Công thức truy hồi tuyến tính thuần nhất với hệ số hằng

Ví dụ 3: Tìm công thức hiển cho nghiệm của

$$\begin{cases} x_n = -3x_{n-1} - 3x_{n-2} - x_{n-3} \\ x_0 = 1, x_1 = -2, x_2 = -1. \end{cases}$$

Phương trình đặc trưng $r^3 + 3r^2 + 3r + 1 = 0$
có nghiệm thực duy nhất $r_0 = -1$ bội 3. Vậy

$$x_n = (\alpha_0 + \alpha_1 n + \alpha_2 n^2)(-1)^n$$

Vì $x_0 = 1, x_1 = -2, x_2 = -1$, ta có

$$\begin{cases} \alpha_0 = 1 \\ \alpha_0 - \alpha_1 - \alpha_2 = -2 \\ \alpha_0 + 2\alpha_1 + 4\alpha_2 = -1 \end{cases} \Leftrightarrow \begin{cases} \alpha_0 = 1 \\ \alpha_1 = 3 \\ \alpha_2 = -2 \end{cases}$$

Công thức hiển:

$$x_n = (1 + 3n - 2n^2)(-1)^n.$$

Đếm bằng phương pháp truy hồi

Sử dụng các công thức truy hồi để giải bài toán đếm

- Thiết lập công thức truy hồi
  - *Lược đồ:* xác định các tham số nào là chỉ số, xây dựng các giá trị đếm dưới dạng công thức truy hồi
- Giải công thức truy hồi tuyến tính
  - Thuần nhất với hệ số hằng
  - Không thuần nhất với hệ số hằng

Công thức truy hồi tuyến tính không thuần nhất với hệ số hằng

Định nghĩa

Một *công thức truy hồi tuyến tính không thuần nhất bậc k với hệ số hằng* có dạng

$$x_n = c_1 x_{n-1} + c_2 x_{n-2} + \ldots + c_k x_{n-k} + F(n),$$

trong đó $c_1, \ldots, c_k$ là các hệ số thực, $c_k \neq 0$, và $F(n)$ là hàm số theo n không đồng nhất bằng 0.

Ghi chú:

- **Tuyến tính:** vế phải là hàm tuyến tính theo k thành phần
- **Không thuần nhất:** xuất hiện thành phần $F(n)$
- **Hệ số hằng:** $c_1, \ldots, c_k$ là các hằng số thực

Ví dụ:

- Bậc 1: $x_n = x_{n-1} + (n+1)$
- Bậc 2: $x_n = x_{n-1} + 2x_{n-2} + 1$

Công thức truy hồi tuyến tính không thuần nhất với hệ số hằng

$$\begin{cases} x_n = c_1 x_{n-1} + c_2 x_{n-2} + \ldots + c_k x_{n-k} + F(n) & (1) \\ x_0 = a_0, x_1 = a_1, \ldots, x_{k-1} = a_{k-1} & (2) \end{cases}$$

Công thức truy hồi thuần nhất liên kết với (1)*:*

$$x_n = c_1 x_{n-1} + c_2 x_{n-2} + \ldots + c_k x_{n-k} \qquad (3)$$

Định lý

Nếu $x_n^{(p)}$ là một nghiệm riêng của (1), thì mọi nghiệm của (1) có dạng

$$x_n = x_n^{(p)} + x_n^{(h)}$$

trong đó $x_n^{(h)}$ là một nghiệm của (3).

Công thức truy hồi tuyến tính không thuần nhất với hệ số hằng

$$x_n = c_1 x_{n-1} + c_2 x_{n-2} + \ldots + c_k x_{n-k} + F(n) \qquad (1)$$

Phương trình đặc trưng (liên kết với phần thuần nhất):

$$r^k - c_1 r^{k-1} - c_2 r^{k-2} - \ldots - c_k = 0 \qquad (4)$$

Định lý (nghiệm riêng trong một số trường hợp đặc biệt)

- Nếu $F(n) = (b_0 + b_1 n + \ldots + b_t n^t)s^n$ với $b_0, b_1, \ldots, b_t, s \in \mathbb{R}$ và s KHÔNG là nghiệm của (4), thì (1) có nghiệm riêng dạng

$$x_n^{(p)} = (p_0 + p_1 n + \ldots + p_t n^t)s^n.$$

- Nếu $F(n) = (b_0 + b_1 n + \ldots + b_t n^t)s^n$ với $b_0, b_1, \ldots, b_t, s \in \mathbb{R}$ và s là nghiệm của (4) với bội m, thì (1) có nghiệm riêng dạng

$$x_n^{(p)} = n^m(p_0 + p_1 n + \ldots + p_t n^t)s^n.$$

Công thức truy hồi tuyến tính không thuần nhất với hệ số hằng

Ví dụ 1: Tìm công thức hiển cho nghiệm của

$$\begin{cases} x_n = x_{n-1} + n & (1) \\ x_0 = 1 & (2) \end{cases}$$

Phương trình đặc trưng $r - 1 = 0$ có nghiệm duy nhất $s = 1$ với bội 1.
Vậy nghiệm của phần thuần nhất có dạng

$$x_n^{(h)} = \alpha \times 1^n = \alpha, \quad \text{với } \alpha \text{ là một hằng số.}$$

Vì $F(n) = n = ns^n$, nghiệm riêng có dạng

$$x_n^{(p)} = n(p_0 + p_1 n)s^n = p_0 n + p_1 n^2.$$

Vậy nghiệm của (1) & (2) có dạng $x_n = x_n^{(p)} + x_n^{(h)} = \alpha + p_0 n + p_1 n^2$.
Thay vào (1) và thu gọn biểu thức thu được, ta có

$$(2p_1 - 1)n + (p_0 - p_1) = 0$$

dẫn đến $2p_1 - 1 = 0$ và $p_0 - p_1 = 0$, vậy $p_0 = p_1 = \frac{1}{2}$.
Từ (2) ta có $\alpha = 1$. Ta thu được công thức hiển

$$x_n = 1 + \frac{1}{2}n + \frac{1}{2}n^2.$$

Công thức truy hồi tuyến tính không thuần nhất với hệ số hằng

Ví dụ 2: Tìm công thức hiển cho nghiệm của

$$\begin{cases} x_n = 2x_{n-1} + 1 \\ x_1 = 1 \end{cases}$$

Ta có thể sử dụng tiếp cận lặp.

$$\begin{aligned} x_n &= 2x_{n-1} + 1 \\ &= 2(2x_{n-2} + 1) + 1 && = 2^2 x_{n-2} + 2 + 1 \\ &= 2^2(2x_{n-3} + 1) + 2 + 1 && = 2^3 x_{n-3} + 2^2 + 2 + 1 \\ &= \dots \\ &= 2^{n-1}x_1 + 2^{n-2} + 2^{n-3} + \dots + 2^2 + 2 + 1 \\ &= 2^{n-1} + 2^{n-2} + 2^{n-3} + \dots + 2^2 + 2 + 1 \\ &= 2^n - 1. \end{aligned}$$

Mục lục

Phát biểu bài toán

Cho trước:

- Các tập hợp $A_1, \ldots, A_m$
- Sơ đồ sắp xếp S
- Các điều kiện $R_1, \ldots, R_n$

Định nghĩa

Mỗi cách sắp xếp các phần tử trong $A_1, \ldots, A_m$

- tuân theo sơ đồ sắp xếp S, và
- thỏa mãn các điều kiện $R_1, \ldots, R_n$

được gọi là một *cấu hình tổ hợp* trên $A_1, \ldots, A_m$

Bài toán: *Liệt kê* tất cả các cấu hình tổ hợp như vậy.

Tiếp cận giải: xây dựng một *thuật toán* với mục đích

- Liệt kê tất cả các cấu hình tổ hợp hợp lệ
- Không cấu hình tổ hợp nào bị lặp lại
- Không cấu hình tổ hợp hợp lệ nào bị bỏ qua

Mục lục

Khái niệm thuật toán

- Thuật ngữ "Thuật toán" (Algorithm) vinh danh al-Khwarizmi[17]
- 825: được viết trong tiếng Ả-rập,
  1230: được sử dụng trong tiếng Anh,
  Cuối thế kỷ 19: có nghĩa như trong tiếng Anh hiện đại

Định nghĩa (không chính thức)

Một thuật toán là một dãy hữu hạn các thao tác đặt chỉnh, có thể cài đặt trên máy tính để giải một lớp các bài toán hoặc thực thi một tính toán.

Đặc trưng:

- Đầu vào
- Đầu ra
- Tính chính xác
- Tính hữu hạn
- Tính duy nhất
- Tính phổ quát

[17] Muhammad ibn Musa al-Khwarizmi (780–850): nhà toán học, nhà thiên văn, nhà địa lý, học giả người Ba Tư trong Thư viện Trí tuệ tại Baghdad, Iraq

Một ví dụ về thuật toán

Bài toán: Tìm giá trị lớn nhất của các số $a, b, c \in \mathbb{R}$ cho trước.

Thuật toán:

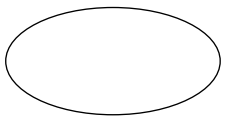
- Bước 1: Gán $x := a$
- Bước 2: Nếu $x < b$, thì gán $x := b$
- Bước 3: Nếu $x < c$, thì gán $x := c$

Với thuật toán này:

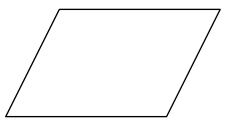
- *Đầu vào:* 3 số thực a, b, c
- *Đầu ra:* x là giá trị lớn nhất của 3 số đầu vào
- *Tính chính xác:* mỗi bước đều được mô tả chính xác
- *Tính hữu hạn:* thuật toán kết thúc sau hữu hạn bước
- *Tính duy nhất:* kết quả tại mỗi bước là duy nhất
- *Tính phổ quát:* áp dụng cho $a, b, c \in \mathbb{R}$ bất kỳ

Sơ đồ khối của thuật toán

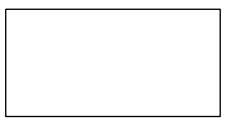
- Một sơ đồ khối là một biểu diễn trực quan các bước của một thuật toán theo thứ tự thực hiện.
- Các ký hiệu cơ bản của sơ đồ khối:



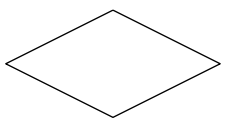
Denote begin or end of program



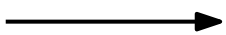
Denote input or output information



Denote computation process



Denote a decision or branching point



Denote direction of flow

Giả mã của thuật toán

Mẫu:

Thuật toán Tên thuật toán
Đầu vào: dữ liệu
Đầu ra: kết quả

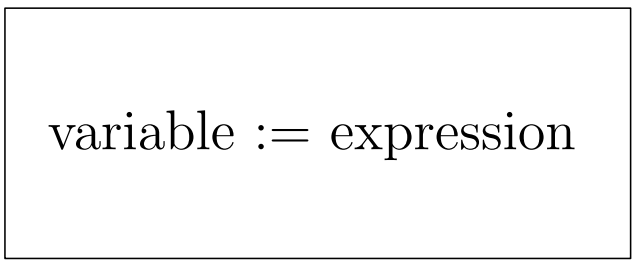
1: Khởi tạo
2: Mô tả các bước của thuật toán
3: **return** kết quả (nếu cần)

Giả mã của thuật toán: cú pháp

- Gán giá trị cho biến:

  biến := biểu thức; *hoặc* biến ← biểu thức;

  Sơ đồ khối:



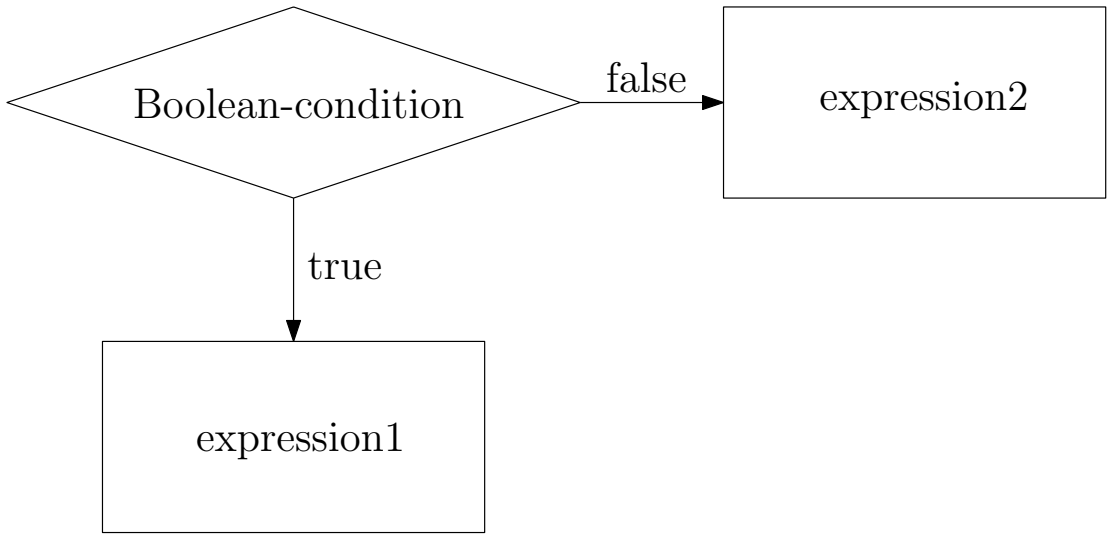
  Ví dụ:

  - $min := a$;
  - $x := \max\{a, b, c\}$;

Giả mã của thuật toán: cú pháp

- Câu lệnh điều kiện:

  if điều kiện Bool **then** biểu thức 1 **else** biểu thức 2 **end if**



Chú ý: phần **else** là tùy chọn

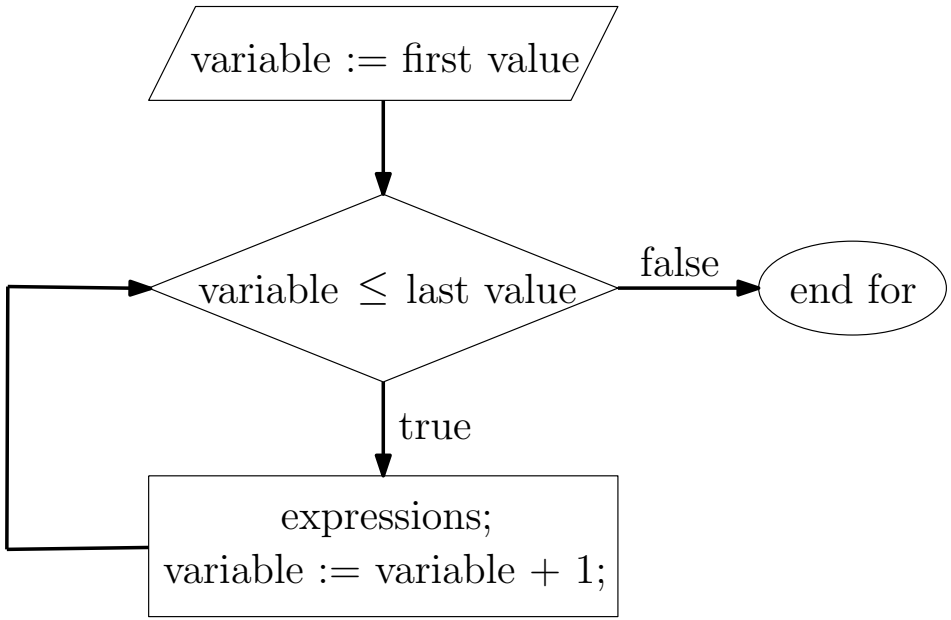
Ví dụ:

- **if** $n \bmod 2 = 0$ **then** $n := n/2$ **else** $n := 3n + 1$; **end if**
- **if** $x = \min\{a, b, c\}$ **then return** x; **end if**

Giả mã của thuật toán: cú pháp

- Vòng lặp for:

  for biến := giá trị đầu tiên **to** giá trị cuối cùng **do** biểu thức **end for**



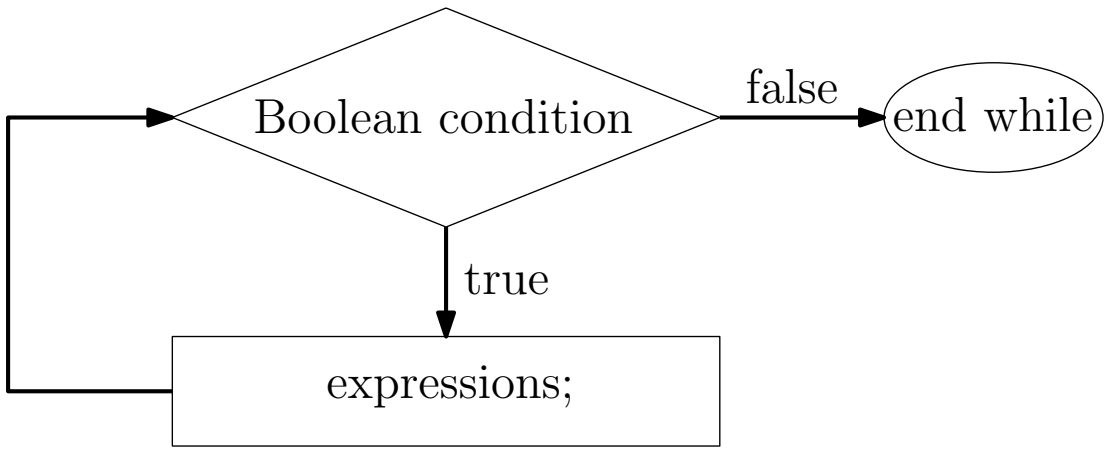
Ví dụ:

max := a_1;
for $i := 2$ **to** n **do**
    if $a_i > max$ **then** $max := a_i$;
end for

Giả mã của thuật toán: cú pháp

- Vòng lặp While:

while điều kiện Bool **do** biểu thức **end while**



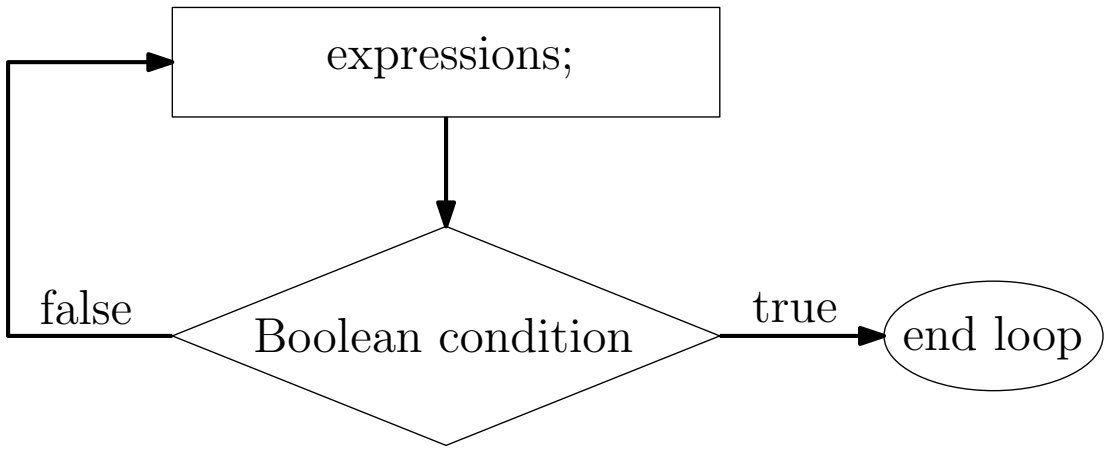
Ví dụ:

$i := 2;$
while $i \leq \sqrt{n}$ **and** $n \bmod i \neq 0$ **do**
    $i := i + 1;$
end while

Giả mã của thuật toán: cú pháp

- Vòng lặp Repeat:

repeat biểu thức **until** điều kiện Bool;



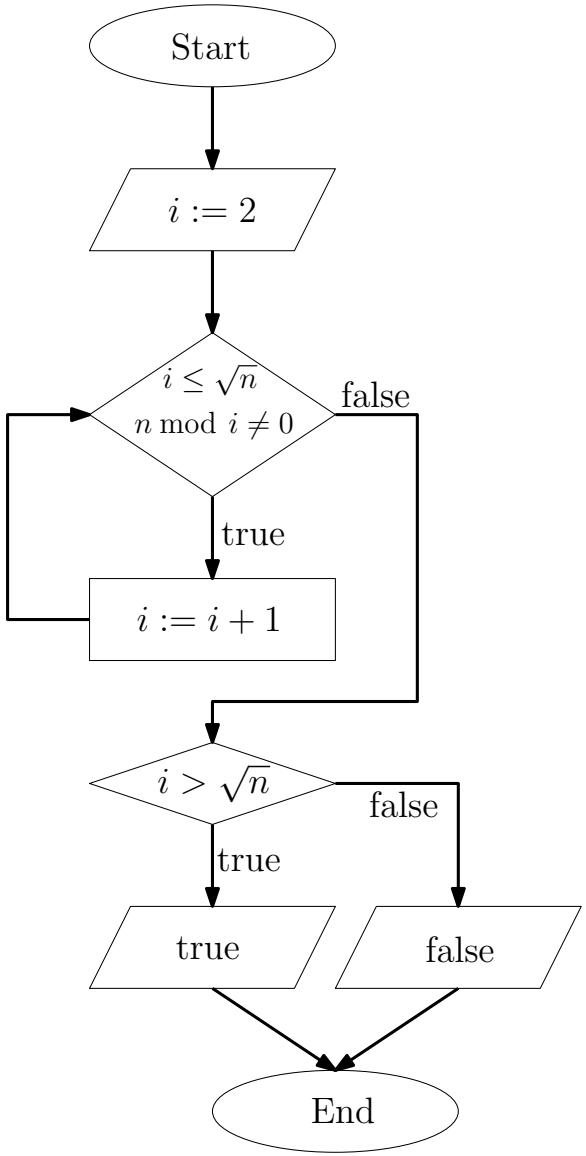
Ví dụ:

$i := 0; s := 0;$
repeat
    $i := i + 1;$
    $s := s + i;$
until $i > n;$

Ví dụ

Thuật toán Kiểm tra tính nguyên tố
Đầu vào: $n \in \mathbb{N}, n \geq 2$
Đầu ra: *true* nếu n nguyên tố, *false* nếu n là hợp số

```
1: i := 2;
2: while i ≤ √n and n mod i ≠ 0 do
3:     i := i + 1;
4: end while
5: if i > √n then
6:     return true;
7: else
8:     return false;
9: end if
```



Khái niệm độ phức tạp tính toán

- Với **dữ liệu đầu vào**, ta thường quan tâm đến cỡ (độ lớn khi lưu trữ trên máy tính)
- Với **thuật toán**, ta thường quan tâm đến khối lượng tính toán, có thể được đo bởi
  - bộ nhớ cần sử dụng trên máy tính
  - thời gian chạy (ta tập trung vào trường hợp này)

Định nghĩa (không chính thức)

Độ phức tạp tính toán của một thuật toán là sự đánh giá tương quan thời gian chạy lâu nhất của thuật toán theo cỡ của dữ liệu đầu vào.

Ký hiệu O

Định nghĩa

Cho $f, g : \mathbb{N} \to \mathbb{R}$.
Ta viết $f(n) = O(g(n))$ nếu tồn tại $M, N > 0$ sao cho

$$|f(n)| \leq M|g(n)| \qquad \text{với mọi } n \geq N,$$

hoặc tương đương

$$\limsup_{n \to +\infty} \frac{|f(n)|}{|g(n)|} < +\infty.$$

Ví dụ:

- $60n^2 + 9n + 10 = O(n^2)$
- $1^k + 2^k + \ldots + n^k = O(n^{k+1})$
- $\log(n!) = O(n \log n)$

Thuật toán thời gian đa thức

Định nghĩa

Một thuật toán được gọi là có *độ phức tạp tính toán đa thức theo thời gian* (hay *có độ phức tạp đa thức*) nếu thời gian chạy thuật toán bị chặn trên bởi một đa thức theo cỡ của dữ liệu đầu vào.

Cụ thể: $T(n) = O(n^k)$ với $k \in \mathbb{N}$,
và $T(n)$ là thời gian chạy lâu nhất của thuật toán trên dữ liệu có cỡ n

Ví dụ: Thuật toán sắp xếp nổi bọt có độ phức tạp $O(n^2)$

Thuật toán Xếp một dãy số theo thứ tự tăng dần

Đầu vào: $a_1, \ldots, a_n \in \mathbb{R}, n \geq 2$

Đầu ra: dãy đã được sắp xếp theo thứ tự tăng dần

```
1: for i := 1 to n − 1 do
2:     for j := i + 1 to n do
3:         if a_i > a_j then
4:             swap(a_i, a_j);
5:         end if
6:     end for
7: end for
```

Thời gian đa thức và thời gian mũ

Độ phức tạp	Thời gian			
	$n = 6$	$n = 12$	$n = 50$	$n = 100$
$O(1)$	10^{-6} sec	10^{-6} sec	10^{-6} sec	10^{-6} sec
$O(n)$	6×10^{-6} sec	10^{-5} sec	5×10^{-5} sec	10^{-4} sec
$O(n^2)$	4×10^{-5} sec	10^{-4} sec	3×10^{-3} sec	0.01 sec
$O(n^3)$	2×10^{-4} sec	2×10^{-3} sec	0.13 sec	1 sec
$O(2^n)$	6×10^{-5} sec	4×10^{-3} sec	36 năm	4×10^6 năm

(10^6 phép tính mỗi giây)

P và NP

Định nghĩa

Một bài toán được gọi là thuộc lớp P nếu nó có thể giải được bởi một thuật toán có độ phức tạp đa thức.

Định nghĩa (không chính thức)

Một *bài toán quyết định* được gọi là *thuộc lớp NP* nếu mọi *câu trả lời yes* đều có thể kiểm tra được bởi một thuật toán có độ phức tạp đa thức.

Ghi chú:

- Bài toán quyết định: câu trả lời hoặc là *yes* hoặc *no*
- NP là viết tắt của *non-deterministic polynomial-time*

Lưu ý: "$P \neq NP$?" là một bài toán triệu đô

Ví dụ:

- Bài toán "Tính $a \times b$ với a, b nguyên tố cho trước" thuộc lớp P
- Bài toán "Xác định xem $c \in \mathbb{N}$ cho trước có phân tích $c = a \times b$ với a, b nguyên tố" thuộc lớp NP

NP-khó và NP-đầy đủ

Định nghĩa

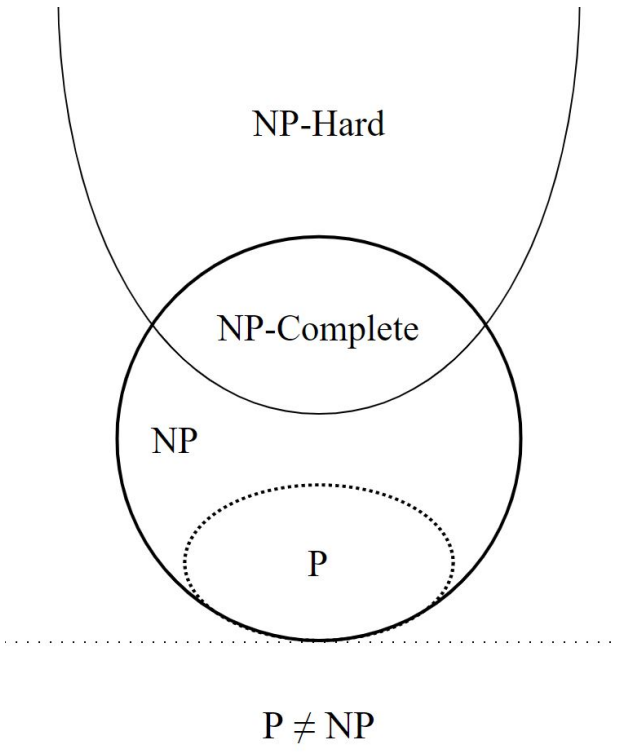
Một bài toán quyết định C được gọi là *thuộc lớp NP-khó* nếu mọi bài toán thuộc lớp NP có thể dẫn về C trong thời gian đa thức.

Nói nôm na:
không dễ hơn bài toán khó nhất trong NP

Định nghĩa

Một bài toán quyết định C được gọi là *thuộc lớp NP-đầy đủ* nếu

- C thuộc lớp NP, và
- mọi bài toán trong lớp NP đều dẫn được về C trong thời gian đa thức.



Mục lục

Phương pháp sinh: Mục đích và phạm vi

Trong trường hợp

- có một thứ tự toàn phần trên tập hợp các cấu hình tổ hợp cần liệt kê, và
- có một thuật toán xây dựng lần lượt các cấu hình tổ hợp,

ta có thể áp dụng phương pháp sinh theo lược đồ sau

Thuật toán Sinh tất cả các cấu hình tổ hợp

1: Xây dựng cấu hình tổ hợp đầu tiên
2: Đặt *currentConfiguration* := cấu hình đầu tiên
3: **while** *currentConfiguration* chưa phải là cấu hình cuối cùng **do**
4: Sinh cấu hình tổ hợp tiếp theo theo thứ tự toàn phần
5: Đặt *currentConfiguration* := cấu hình vừa được sinh
6: **end while**

Ví dụ 1

Bài toán: Liệt kê các xâu nhị phân độ dài n.

Phân tích:

- Mỗi xâu $\mathbf{b} = b_1 b_2 \dots b_n \in \{0,1\}^n$ tương ứng 1-1 với một số tự nhiên

$$p(\mathbf{b}) = b_1 2^{n-1} + b_2 2^{n-2} + \dots + b_n 2^0$$

- Thứ tự toàn phần $\prec$ xác định bởi

$$\mathbf{b} \prec \mathbf{b}' \quad \Leftrightarrow \quad p(\mathbf{b}) < p(\mathbf{b}')$$

$\mathbf{b}$	$p(\mathbf{b})$
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

- Xâu đầu tiên: $00\dots0$
- Quy tắc sinh xâu nhị phân tiếp theo:
  - Cho trước xâu nhị phân $b_1 b_2 \dots b_n$
  - Tìm chỉ số i lớn nhất thỏa mãn $b_i = 0$
  - Đặt $b_i = 1$ và $b_j = 0$ với mọi $j > i$

Ví dụ 1

Thuật toán Sinh tất cả xâu nhị phân độ dài n

1: Đặt $b_i = 0$ với mọi $i = 1, \ldots, n$ và in $b_1 b_2 \ldots b_n$;
2: stop := false;
3: **while** stop = false **do**
4: $i := n$;
5: **while** $i \geq 1$ **and** $b_i = 1$ **do**
6: $b_i := 0$;
7: $i := i - 1$;
8: **end while**
9: **if** $i = 0$ **then**
10: stop := true
11: **else**
12: $b_i := 1$;
13: In $b_1 b_2 \ldots b_n$;
14: **end if**
15: **end while**

Ví dụ 2

Bài toán: Liệt kê mọi tập con gồm m phần tử của $X = \{1, \ldots, n\}$.
Phân tích:

- Mỗi tập con như vậy được biểu diễn bởi một bộ có thứ tự

  $$\mathbf{a} = (a_1, \ldots, a_m) \quad \text{thỏa mãn} \quad 1 \le a_1 < a_2 < \ldots < a_m \le n$$

- Thứ tự toàn phần $\prec$ xác định bởi thứ tự từ điển

  $$\mathbf{a} \prec \mathbf{a}' \quad \Leftrightarrow \quad \exists k \in \{1, \ldots, m\} \text{ s.c. } a_1 = a'_1, \ldots, a_{k-1} = a'_{k-1}, a_k < a'_k$$

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

- Bộ đầu tiên: $(1, 2, \ldots, m)$
- Quy tắc sinh bộ tiếp theo:
  - Cho trước bộ $(a_1, \ldots, a_m)$
  - Tìm chỉ số i lớn nhất thỏa mãn $a_i \neq n - m + i$
  - Đặt $a_i = a_i + 1$ và $a_j = a_i + j - i$ với mọi $j > i$

Ví dụ 2

Thuật toán Sinh tất cả tập con m phần tử của $X = \{1, \ldots, n\}$

1: Đặt $a_i = i$ với mọi $i = 1, \ldots, m$ và in $(a_1, \ldots, a_m)$;
2: stop := false;
3: **while** stop = false **do**
4: $i := m$;
5: **while** $a_i = n - m + i$ **do**
6: $i := i - 1$;
7: **end while**
8: **if** $i = 0$ **then**
9: stop := true
10: **else**
11: $a_i := a_i + 1$;
12: Đặt $a_j := a_i + j - i$ với mọi $j = i + 1, \ldots, m$;
13: In $(a_1, \ldots, a_m)$;
14: **end if**
15: **end while**

Mục lục

Phương pháp quay lui: Mục đích và phạm vi

Trong trường hợp

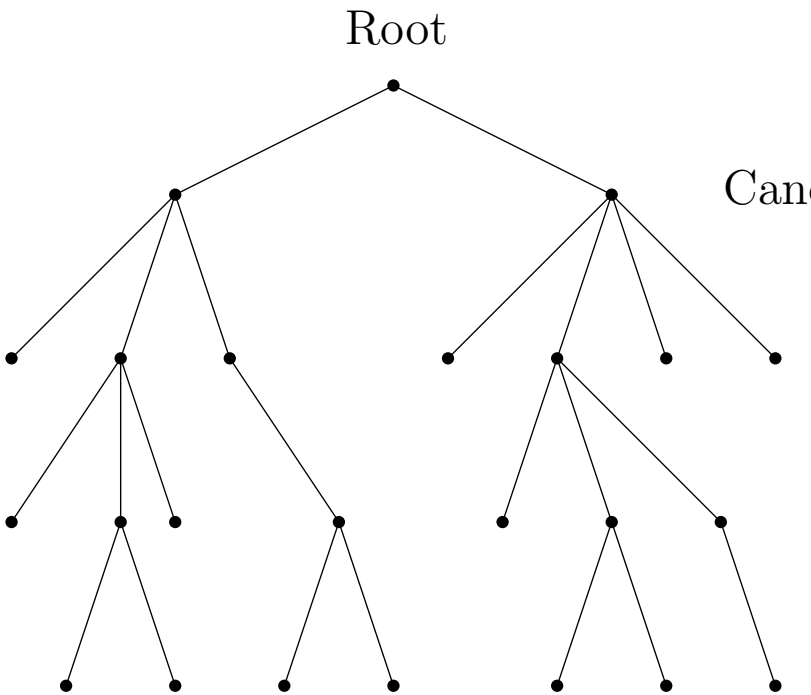
- mỗi cấu hình tổ hợp đều có thể xây dựng từng phần,
- mỗi phần có thể xây dựng bằng cách kiểm tra các khả năng,

ta có thể áp dụng phương pháp quay lui theo lược đồ đệ quy sau

Thuật toán constructComponent(i)

```
1: for mỗi j là một khả năng cho thành phần i do
2:     if j được chấp nhận then
3:         if i là thành phần cuối then
4:             In cấu hình;
5:         else
6:             constructComponent(i + 1);
7:         end if
8:     end if
9: end for
```

Cây tìm kiếm lời giải



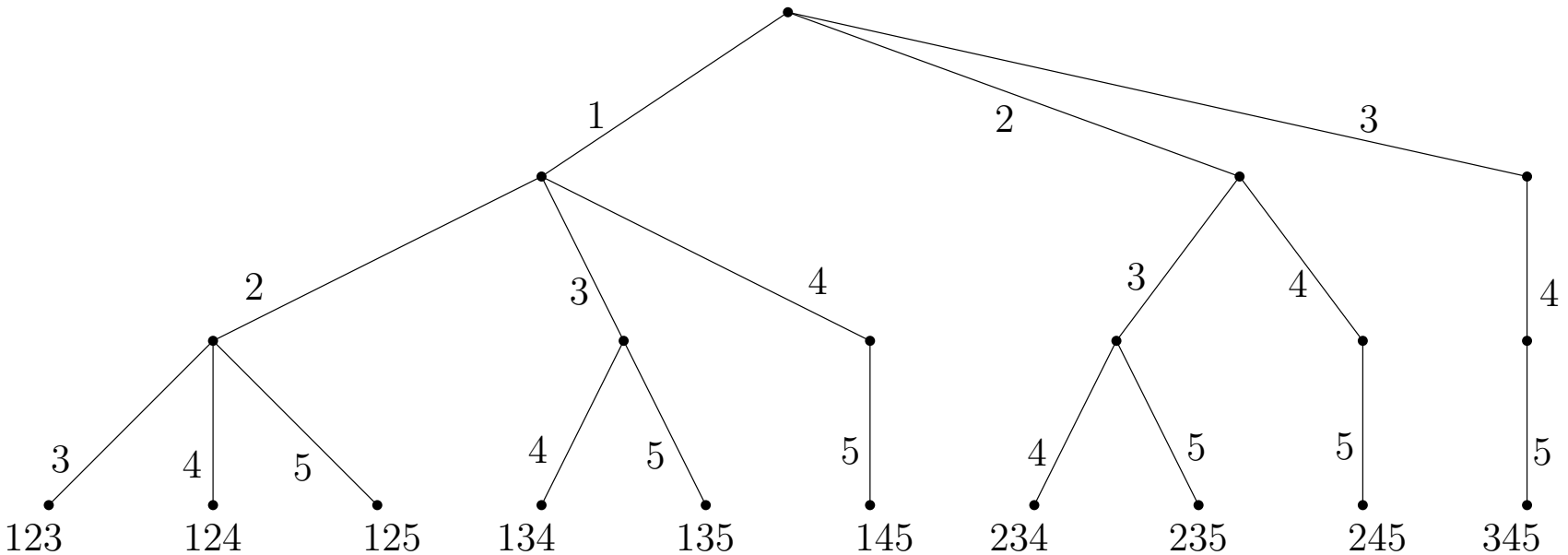
Candidates for 1st component

Candidates for 2nd component
given chosen candidate of 1st component

⋮

Ví dụ 1

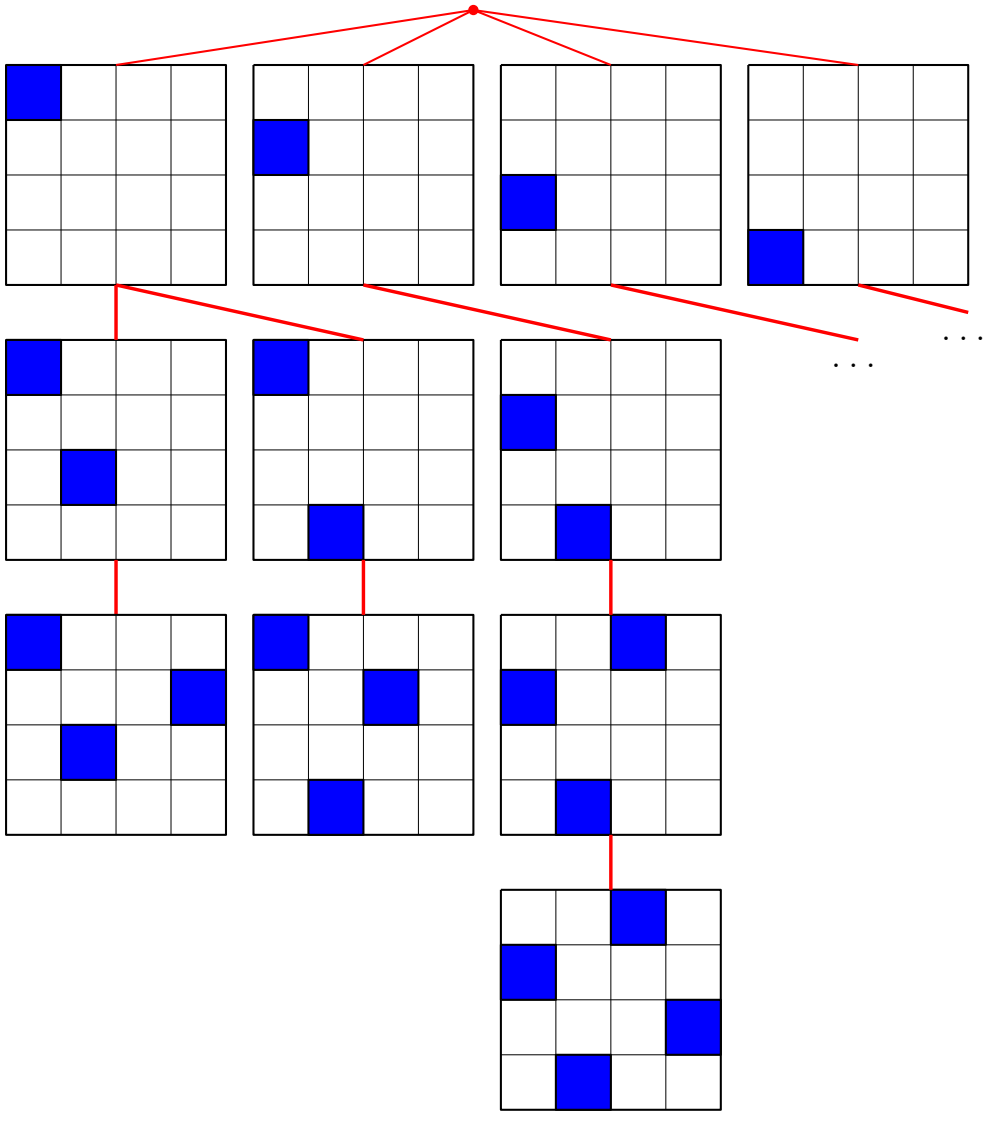
Bài toán: Liệt kê mọi tổ hợp chập 3 phần tử của $X = \{1, \ldots, 5\}$.



Ví dụ 2

Bài toán
***n* quân hậu:**

Đặt n quân hậu lên
một bàn cờ $n \times n$
sao cho không có
2 quân nào chiếu nhau



Mục lục

Phát biểu bài toán

Cho trước:

- Các tập hợp $A_1, \ldots, A_m$
- Sơ đồ sắp xếp S
- Các điều kiện $R_1, \ldots, R_n$

Định nghĩa

Mỗi cách sắp xếp các phần tử trong $A_1, \ldots, A_m$

- tuân theo sơ đồ sắp xếp S, và
- thỏa mãn các điều kiện $R_1, \ldots, R_n$

được gọi là một *cấu hình tổ hợp* trên $A_1, \ldots, A_m$

Bài toán: Trong tất cả các cấu hình tổ hợp hợp lệ, tìm cấu hình tổ hợp *tốt nhất* theo nghĩa nào đó

$$\min \mid \max \qquad f(x) \qquad \text{v.đ.k.} \qquad x \in C$$

với C là tập hợp các cấu hình tổ hợp hợp lệ

Ví dụ 1: Bài toán người đưa hàng (TSP)

Cho trước một danh sách các thành phố và khoảng cách giữa các thành phố, tìm chu trình ngắn nhất đi qua tất cả các thành phố và quay lại thành phố ban đầu.

Lời giải TSP cho 49 thành phố Mỹ. *Newsweek*, 26.07.1954

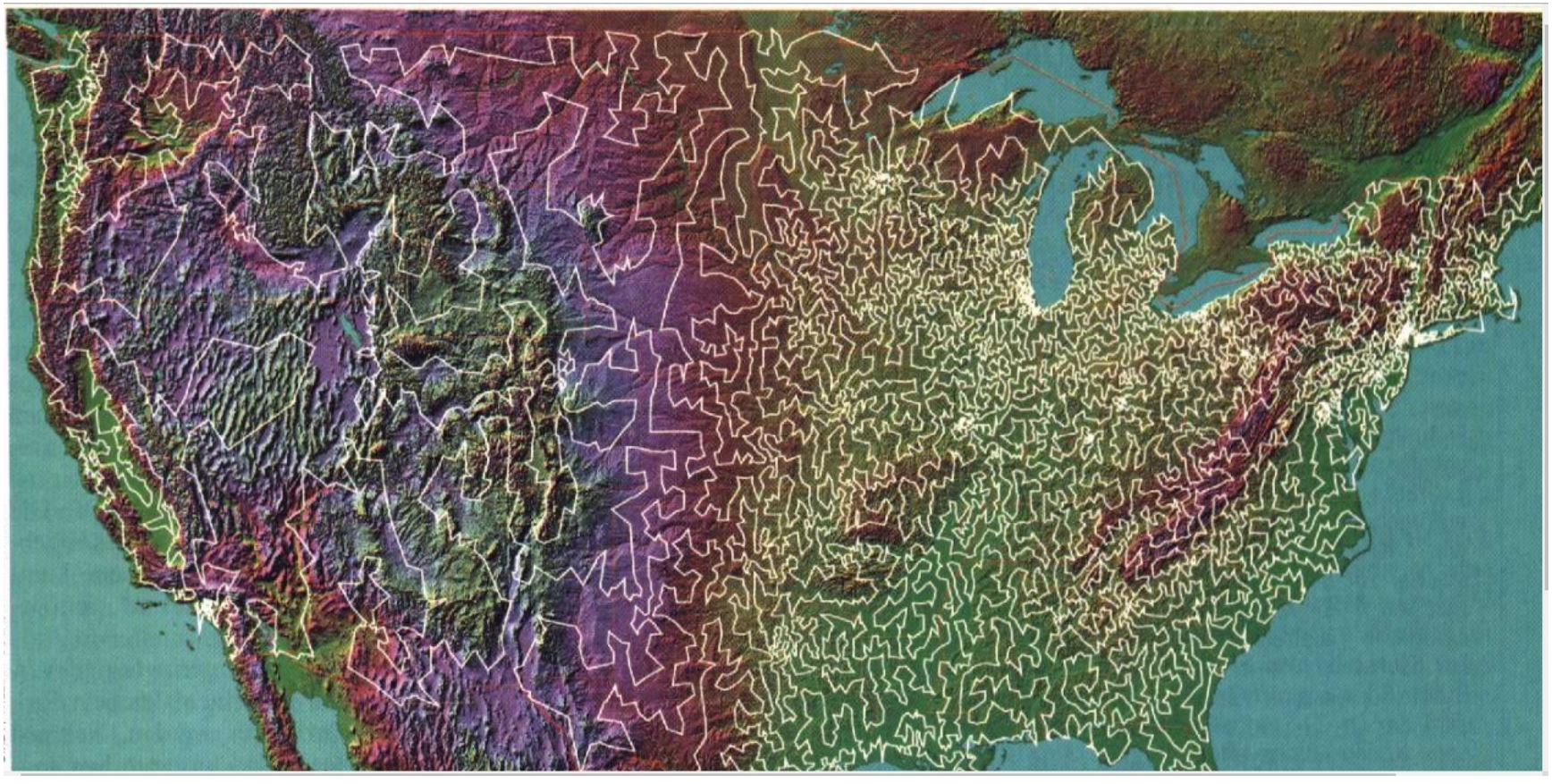
TSP: Lý do nghiên cứu

- Được phát biểu từ 1930
- Một trong những bài toán tối ưu được nghiên cứu nhiều nhất
- Phạm vi ứng dụng rộng rãi
- Bài toán mở trong lý thuyết độ phức tạp tính toán
  - TSP có thể giải trong thời gian đa thức $\Leftrightarrow$ $P = NP$
- Liệt kê tất cả các chu trình, sau đó tìm chu trình ngắn nhất?
  - Không thể: $n!$ chu trình với n thành phố

Các cột mốc trong việc giải TSP

Year	Research Team	Size of Instance
1954	G. Dantzig, R. Fulkerson, and S. Johnson	49 cities
1971	M. Held and R.M. Karp	64 cities
1975	P.M. Camerini, L. Fratta, and F. Maffioli	67 cities
1977	M. Grötschel	120 cities
1980	H. Crowder and M.W. Padberg	318 cities
1987	M. Padberg and G. Rinaldi	532 cities
1987	M. Grötschel and O. Holland	666 cities
1987	M. Padberg and G. Rinaldi	2,392 cities
1994	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	7,397 cities
1998	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	13,509 cities
2001	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	15,112 cities
2004	D. Applegate, R. Bixby, V. Chvátal, W. Cook, and K. Helsgaun	24,978 cities

American tour: 13509 thành phố



Lời giải TSP cho 13509 thành phố Mỹ
(*Spektrum der Wissenschaft*, 04.1999)

Cập nhật: 115475 thành phố Mỹ, 07.2012

Sweden tour: 24978 thành phố

http://www.math.uwaterloo.ca/tsp/sweden/index.html

TSP: các thành tựu tính toán gần đây

- TSP thế giới: 1,904,711 thành phố
  - Cận dưới tốt nhất hiện tại:

    7 512 218 268 km

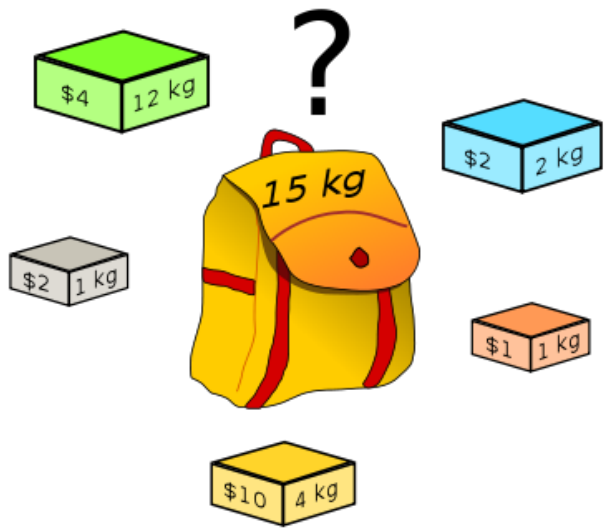
  - Cận trên tốt nhất hiện tại:

    7 515 772 107 km

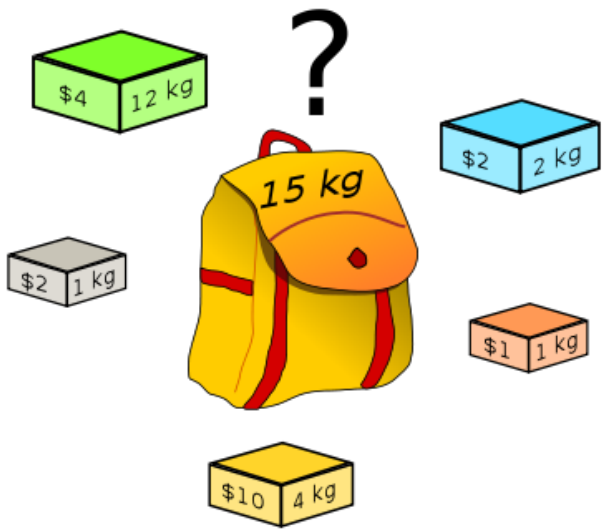
    tìm được bởi Keld Helsgaun vào 13.03.2018, được chứng minh là không dài hơn 0.0474% so với độ dài chu trình tối ưu

Ví dụ 2: Bài toán Knapsack 1-D

- Cho trước một tập hợp các đồ vật $\{1, \ldots, n\}$
- Đồ vật i có khối lượng w_i và giá trị c_i
- Cho trước một túi có thể đựng khối lượng không quá W
- Cần chọn các đồ vật nào đặt vào trong túi
  sao cho tổng khối lượng không quá W
  và tổng giá trị là lớn nhất



Ví dụ 2: Bài toán Knapsack 1-D



Biến: $x_i = 1$ nếu đồ vật i được chọn, $= 0$ nếu ngược lại

Mô hình:

$$\max \quad \sum_{i=1}^{n} c_i x_i$$

$$\text{v.đ.k.} \quad \sum_{i=1}^{n} w_i x_i \leq W$$

$$x_i \in \{0, 1\} \qquad \forall i = 1, \ldots, n$$

Bài toán Knapsack: Lý do nghiên cứu

- Được phát biểu từ 1897
- Một trong những bài toán tối ưu được nghiên cứu nhiều nhất
- Phạm vi ứng dụng rộng rãi
- Phiên bản quyết định là NP-đầy đủ,
  nhưng có thuật toán xấp xỉ trong thời gian đa thức

Trân trọng cảm ơn!