

# PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN (Design and Analysis of Algorithms)

**L/O/G/O**

GV: HUỖNH THỊ THANH THƯỜNG

Email: [hh.thanhthuong@gmail.com](mailto:hh.thanhthuong@gmail.com)

[thuonghtt@uit.edu.vn](mailto:thuonghtt@uit.edu.vn)

## CHƯƠNG 3

# THIẾT KẾ THUẬT TOÁN

Algorithm Design

# Bài toán tối ưu tổ hợp

❖ Có dạng tổng quát:

Cho hàm  $f(X)$  là hàm mục tiêu xác định trên 1 tập hữu hạn các phần tử  $D$  (tập các phương án)

Mỗi  $X \in D$  có dạng  $X = (X_1, X_2, \dots, X_n)$  gọi là 1 phương án,  $X_i \in P$  (tập các biến)

Cần tìm 1 phương án  $X$  “chấp nhận được” (thỏa mọi ràng buộc) sao cho  $f(X)$  đạt min (max) →  
**phương án tối ưu**

# Bài toán tối ưu tổ hợp

## ❖ Cách giải quyết

### ▪ Vét cạn

Tuần tự xét tất cả các khả năng (phương án) có thể có cho đến khi gặp giải pháp cho vấn đề cần giải quyết → **thời gian mũ**

### ▪ Các thuật toán Quy hoạch tuyến tính (Toán học: ngành tối ưu)

### ▪ Tối ưu cục bộ : phương pháp tham lam

# Bài toán tối ưu tổ hợp

## ❖ Cách giải quyết

- Các thuật toán Quy hoạch tuyến tính  
(Toán học: ngành tối ưu)

Các cách giải đơn giản:

- Giải bằng hình học (trường hợp 2 biến)
- Thuật toán Fourier-Motzkin

Phức tạp hơn: Phương pháp đơn hình, phương pháp trọng tâm, ...

# Giải bài toán tối ưu tổ hợp

## ❖ Ví dụ:

Một công ty điện tử sản xuất 2 kiểu radio trên 2 dây chuyền độc lập. Công suất của dây chuyền 1 là 60 radio/ngày và dây chuyền 2 là 75 radio/ngày. Để sản xuất 1 chiếc radio kiểu 1 cần 10 linh kiện điện tử E và 1 chiếc radio kiểu 2 cần 8 linh kiện này. Số linh kiện này được cung cấp mỗi ngày không quá 800. Tiền lãi khi bán 1 radio kiểu 1 là 30USD và kiểu 2 là 20USD. Xác định phương án sản xuất cho lãi nhiều nhất trong ngày.

Giải bằng hình học và bằng phương pháp Fourier-Motzkin.

Đầu tiên phải mô hình hóa bài toán:

Gọi  $x_1, x_2$  là số lượng radio kiểu 1 và kiểu 2 sản xuất trong 1 ngày

Hàm mục tiêu:  $\max z = 30x_1 + 20x_2$

Các ràng buộc:  $x_1 \leq 60, x_2 \leq 75, 10x_1 + 8x_2 \leq 800, x_1 \geq 0, x_2 \geq 0$ .

# Giải bài toán tối ưu tổ hợp

## ❖ Giải bằng hình học

Đường mức  $30x_1 + 20x_2 = \alpha$

Các ràng buộc được đánh số thứ

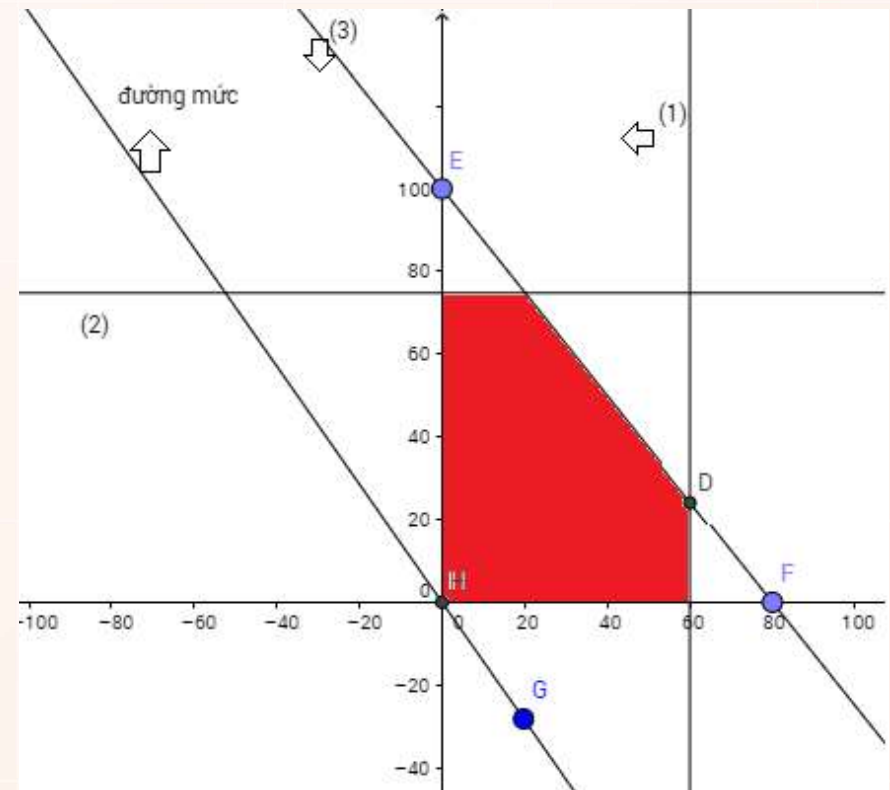
tự:

- (1)  $x_1 \leq 60$
- (2)  $x_2 \leq 75$
- (3)  $10x_1 + 8x_2 \leq 800$
- (4)  $x_1 \geq 0$
- (5)  $x_2 \geq 0$

Điểm cuối cùng mà đường mức còn cắt miền chấp nhận được là điểm D.

D là giao điểm của (1) và (3), giải hệ phương trình ta được tọa độ của D là (60, 25)

Vậy phương án tối ưu là  $x_1 = 60$  và  $x_2 = 25$  với lãi suất cao nhất 2300 USD



# Giải bài toán tối ưu tổ hợp

## ❖ Giải bằng thuật toán Fourier-Motzkin

Các bất phương trình được đánh số thứ tự:

- (1)  $x_1 \leq 60$
- (2)  $x_2 \leq 75$
- (3)  $10x_1 + 8x_2 \leq 800$
- (4)  $x_1 \geq 0$
- (5)  $x_2 \geq 0$
- (6)  $z \leq 30x_1 + 20x_2$  (viết lại hàm mục tiêu)

Trên lớp GV hướng dẫn giải bằng cách khử  $x_2$  trước,  
trong slide GV hướng dẫn lại theo cách khử  $x_1$  trước



# Giải bài toán tối ưu tổ hợp

## ❖ Giải bằng thuật toán Fourier-Motzkin

- (1)  $x_1 \leq 60$
- (2)  $x_2 \leq 75$
- (3)  $10x_1 + 8x_2 \leq 800$
- (4)  $x_1 \geq 0$
- (5)  $x_2 \geq 0$
- (6)  $z \leq 30x_1 + 20x_2$  (viết lại hàm mục tiêu)

Khử  $x_1$  trước:

B1: chuyển  $x_1$  về 1 vế và viết lại các bất đẳng thức sao cho cùng chiều

- (1)  $x_1 \leq 60$
- (2)  $x_2 \leq 75$
- (3)  $x_1 \leq (800 - 8x_2)/10$
- (4)  $0 \leq x_1$
- (5)  $0 \leq x_2$
- (6)  $(z - 20x_2) / 30 \leq x_1$

B2: Ghép từng BĐT dạng  $x_1 \leq \dots$  với các BĐT dạng  $\dots \leq x_1$

- $0 \leq 60$  (hiển nhiên)
- $x_2 \leq 75$
- $0 \leq (800 - 8x_2)/10$
- $0 \leq x_2$
- $(z - 20x_2) / 30 \leq 60$
- $(z - 20x_2) / 30 \leq (800 - 8x_2)/10$

# Giải bài toán tối ưu tổ hợp

Tương tự: Khử  $x_2$

B1: Đưa  $x_2$  về 1 vế

- (1)  $x_2 \leq 75$
- (2)  $x_2 \leq 100$  (có thể bỏ cho gọn)
- (3)  $0 \leq x_2$
- (4)  $(z - 1800)/20 \leq x_2$
- (5)  $x_2 \leq (2400 - z)/4$

B2: Ghép các BĐT

$$\begin{aligned}0 &\leq 75 \\0 &\leq (2400 - z)/4 \\(z - 1800)/20 &\leq 75 \\(z - 1800)/20 &\leq (2400 - z)/4\end{aligned}$$

Rút gọn:

$$\begin{aligned}z &\leq 2400 \\z &\leq 3300 \\z &\leq 2300\end{aligned}$$

Kết luận:  $z \leq 2300$  nên  $\max z = 2300$

Thế lại ta được:

$$\begin{aligned}60 &\leq x_1 \leq 60 \text{ nên } x_1 = 60 \\25 &\leq x_2 \leq 25 \text{ nên } x_2 = 25\end{aligned}$$

# BONUS THỰC HÀNH

Bonus vào Cột điểm thực hành (Thêm 1 cột điểm TH 10 điểm khi tính TBC):

Cài đặt thuật toán **Fourier-Motzkin**

- Dùng NNLT nào cũng được (gv gợi ý dùng Mapple vì ẻm hỗ trợ xử lý tính toán mạnh mẽ)
- Thuật toán này chỉ giải được 1 số dạng đặc biệt của bài toán tối ưu, chứ không phải giải được mọi loại bài toán.
- SV chỉ cần giải quyết trường hợp 2, 3 biến
- 2 test case bắt buộc có là bài “Sản xuất radio” và “Sản xuất sơn nội thất/ngoài trời”, SV có thể tự tạo thêm 2-3 test case khác nữa.
- Nộp bài gồm: 1) source code, 2) các testcase kèm mô tả bài toán (nếu có), 3) video ghi lại quá trình chạy thử nghiệm, 4) Không bắt buộc, bài giải thủ công bằng tay để đối chiếu các đáp án
- SV ko cần mất thời gian search google vì có search cũng chẳng có kết quả gì đâu, tự cài đặt theo cách hiểu và theo kỹ năng lập trình của mình

# Nội dung

- ❖ Phương pháp chia để trị, giảm để trị, biến đổi để trị
- ❖ Phương pháp tham lam
- ❖ **Phương pháp quay lui**  
**(Backtracking method)**
- ❖ Phương pháp quy hoạch động
- ❖ Các phương pháp khác

# Phương pháp quay lui

## HÌNH ẢNH MINH HỌA

- Tại mỗi bước có nhiều lựa chọn, ta chọn thử 1 bước để đi tiếp.
- Phân tích đi xuống cho tới khi đạt điểm dừng và quay lui trở lại theo con đường đã đi qua



# Phương pháp quay lui

## ĐẶC TRƯNG:

- ❖ Vết cặn
- ❖ Lời giải: được cấu thành bởi nhiều **bước giải**
- ❖ Quá trình tìm lời giải:
  - Xây dựng từng bước giải thành phần
  - Hoàn toàn dựa trên các phép thử (phương pháp thử và sai - try and error)

# Phương pháp quay lui

## MÔ HÌNH HÓA:

❖ Ý tưởng:

Lời giải của bài toán là một bộ  $S = \langle S_1, S_2, \dots, S_n \rangle$  phải thỏa mãn điều kiện nào đó.

Tại bước thứ  $i \equiv$  xây dựng bước giải  $S_i$  :

- Đã xây dựng xong các thành phần  $S_1, \dots, S_{i-1}$
- **Xây dựng  $S_i$**  bằng cách lần lượt thử tất cả các khả năng mà  $S_i$  có thể chọn,

# Phương pháp quay lui

## ❖ MÔ HÌNH

try (i) =

```
{    for (j = 1 --> k) // k: số khả năng mà  $S_i$  có thể chọn
    if (khả năng j chấp nhận được)
    {    Xác định  $S_i$  theo j;
        Ghi nhận trạng thái mới của bài toán (nếu có);
        if (i < n)
            try (i+1); // tiến hành bước i+1 để xác định tiếp  $S_{i+1}$ 
        else
            Xuất lời giải  $S = (S_1, S_2, \dots, S_n)$ ;
        Trả lại trạng thái cũ của bài toán;
    }
```



# Phương pháp quay lui

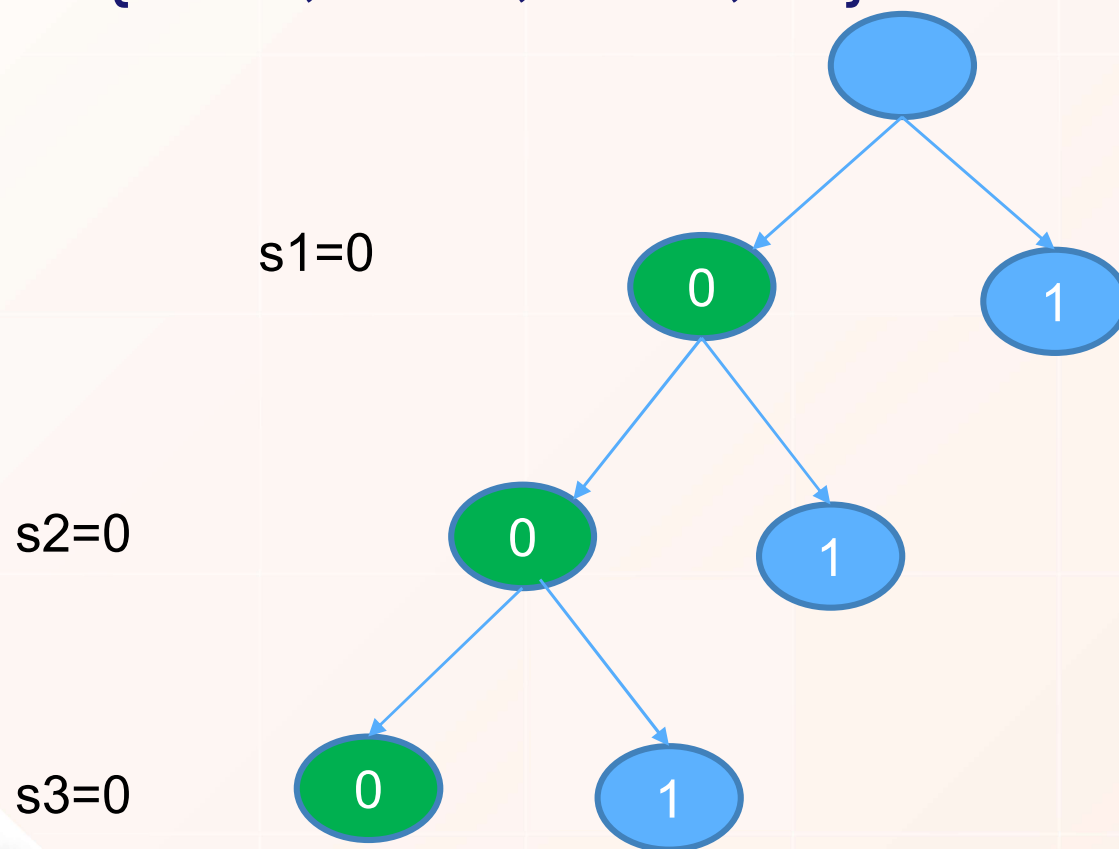
❖ Ví dụ 1: Bài toán liệt kê tất cả các dãy số nhị phân có độ dài  $n$  cho trước

○ Mẫu: nếu  $n = 4$ , ta có các lời giải sau:

{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111  
1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111}

# Phương pháp quay lui

- Mẫu: nếu  $n = 4$ , ta có các lời giải sau:  
 $\{0000, 0001, 0010, \dots\}$



# Phương pháp quay lui

## ❖ Ví dụ 1:

Lời giải  $s = \langle S_1, S_2, \dots, S_n \rangle$ , mỗi  $S_i$  thuộc  $\{0, 1\}$

```
try (i) =  
{ for (j = 0; j <= 1; j++) //  $S_i$  chỉ có 2 khả năng để chọn là 0, 1  
    // giá trị 0, 1, mặc nhiên chấp nhận được nên không cần kiểm tra điều kiện if ( $S_i$  chấp  
    nhận được khả năng j)  
    {  $S[i] = j$ ; // Xác định  $S_i$  theo j, lựa chọn j cho  $S_i$   
      if (i < n)  
        try (i+1);  
      else  
        Xuất lời giải S;  
      // VD này không có bước trả về trạng thái cũ  
    }  
}
```

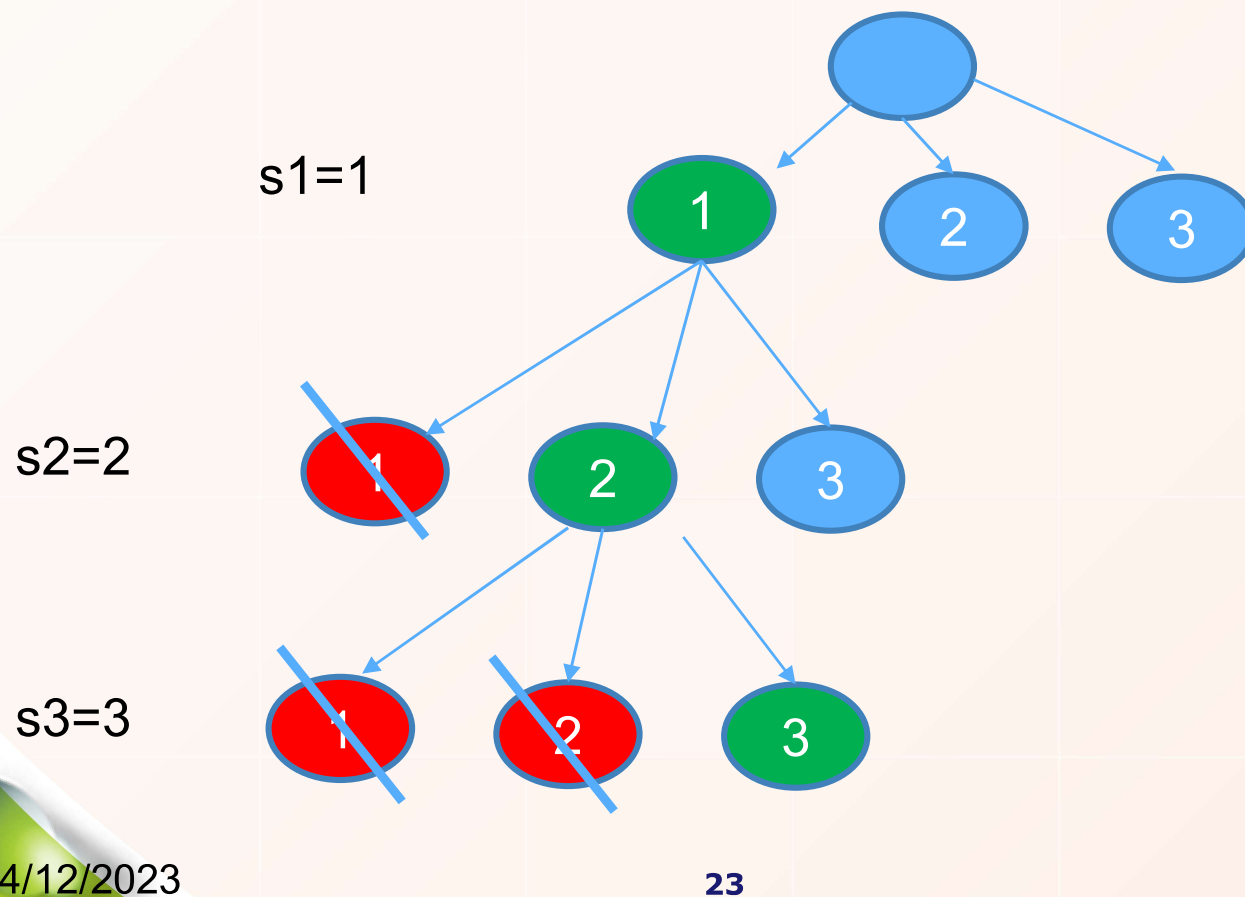
# Phương pháp quay lui

❖ Ví dụ 2: Bài toán liệt kê tất cả các hoán vị của  $n$  số nguyên dương đầu tiên

○ Mẫu: nếu  $n = 3$ , ta có các lời giải sau:  
 $\{123, 132, 213, 231, 312, 321\}$

# Phương pháp quay lui

- Mẫu: nếu  $n = 3$ , ta có các lời giải sau:  
 $\{123, 132, 213, 231, 312, 321\}$



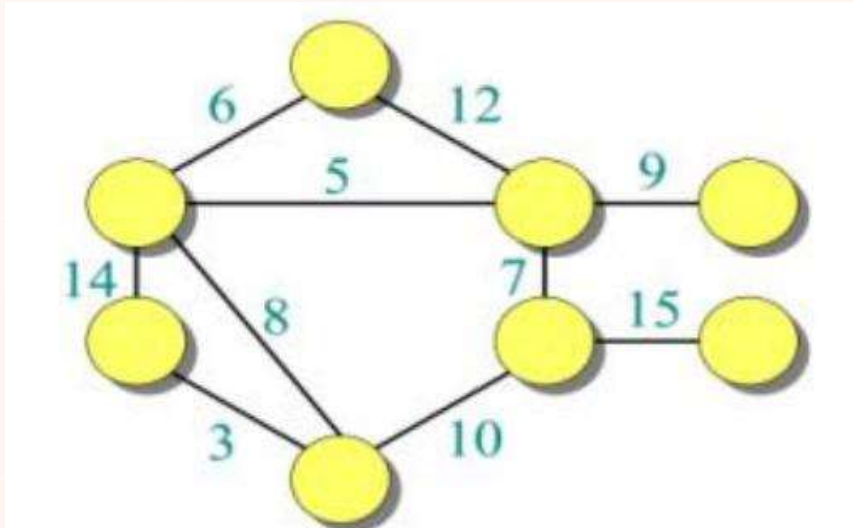
# Phương pháp quay lui

## ❖ Ví dụ 3: Giải đề thi

**(Câu 4.1)** Cho  $G = (V, E)$  là một đồ thị vô hướng liên thông, trong đó  $V$  là tập đỉnh,  $E$  là tập cạnh và các cạnh đều có trọng số. Cây  $T = (V, F)$  với  $F \subset E$  được gọi là cây khung của  $G$ . Cây không có chu trình và có  $n - 1$  cạnh. Cây khung ngắn nhất hay còn gọi là cây khung tối thiểu là cây khung của  $G$  có tổng độ dài (trọng số) các cạnh nhỏ nhất. Tìm cây khung tối thiểu của  $G$ .

Yêu cầu :

- Hãy thiết kế một thuật toán theo chiến lược **“Quay lui (Backtracking)”** để giải bài toán trên (trình bày dưới dạng mã giả và có chú thích cho người đọc dễ hiểu).

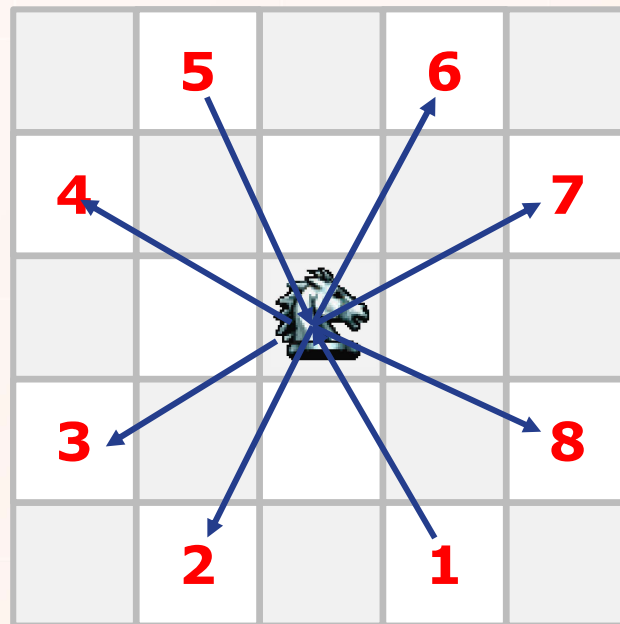


**ĐỀ THI CUỐI HỌC KỲ I (2022-2023)**  
**MÔN: CS112 - PHÂN TÍCH VÀ THIẾT**  
**KẾ THUẬT TOÁN**  
*Thời gian: 90 phút*

# Bài tập tại lớp

## ❖ Ví dụ 4: Bài toán con mã đi tuần

- Cho bàn cờ vua kích thước 8x8 (64 ô)
- Hãy đi con mã 64 nước sao cho mỗi ô chỉ đi qua 1 lần (xuất phát từ ô bất kỳ) theo luật:



BONUS: SV XÂY DỰNG 1  
GAME MINI CÓ GIAO DIỆN  
(copy source code trên mạng  
là không hợp lệ)



# Câu hỏi định hướng

Cho trước 1 bàn cờ có kích thước  $n \times n$  (Ví dụ  $8 \times 8$ )

Input:

- $(x_0, y_0)$  cho biết vị trí đầu tiên đặt con mã lên bàn cờ

Output:

- 1 số nguyên không âm cho biết số hành trình hợp lệ mà con mã có thể đi được
- Liệt kê tất cả các hành trình

Mỗi hành trình có thể được biểu diễn dưới dạng 1 ma trận  $a(n \times n)$ , trong đó

$a_{ij} = 1$  số nguyên (từ 1-64) cho biết con mã đã đi đến ô  $(i, j)$  tại bước đi thứ mấy trong hành trình

Mã giả:

$\text{Madituan}(i, \dots)$  = Xác định bước đi thứ  $i$  của con mã trong hành trình

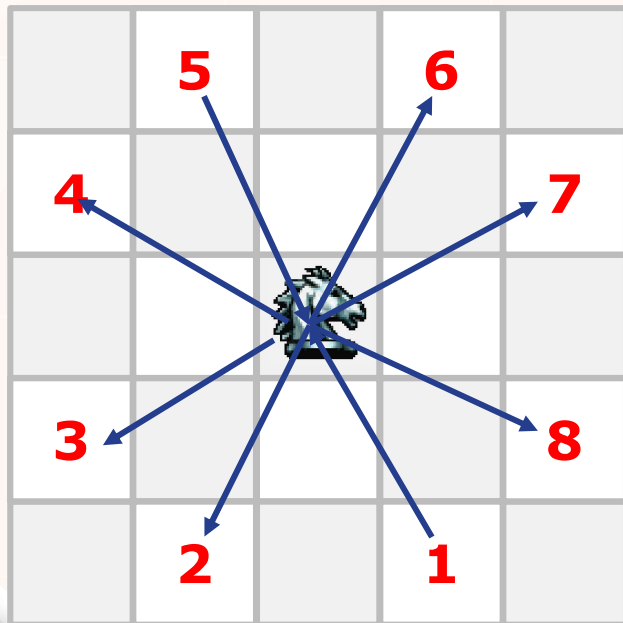
```
{
  for(j=1->8) // có 8 vị trí có thể xem xét lựa chọn ở mỗi bước
  {
    Làm sao cho máy tính hiểu được j=1,2,...8 tương ứng với vị trí nào trên bàn cờ
    if(khi nào thì 1 bước đi là hợp lệ)
    {
      - Lưu lại lựa chọn cho bước đi thứ i
      - Cập nhật trạng thái mới của bài toán (nếu có)
      - if(làm sao biết được là đã xong hành trình)
        + Lưu lại và xuất hành trình như thế nào
      else
        + Gọi đệ quy để tìm bước đi tiếp theo
      - Trả lại trạng thái cũ cho bài toán trước khi quay lại for để thử qua khả năng khác
    }
  }
}
```



# BONUS THỰC HÀNH

## ❖ Ví dụ 4: Bài toán con mã đi tuần

- Cho bàn cờ vua kích thước 8x8 (64 ô)
- Hãy đi con mã 64 nước sao cho mỗi ô chỉ đi qua 1 lần (xuất phát từ ô bất kỳ) theo luật:



**SV XÂY DỰNG 1 GAME MINI CÓ GIAO DIỆN**

(copy source code trên mạng là không hợp lệ)