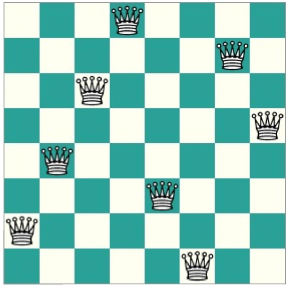
**Bài 15: Thiết kế thuật toán giải bài toán xếp tám hậu theo phương pháp quay lui với các công việc sau:**

**1. Nêu bài toán;**



Tám quân hậu được đặt trên bàn cờ vua sao cho chúng không ăn nhau

**2. Mô tả chi tiết thuật toán;**

Mấu chốt của thuật toán rõ ràng là xét xem có thể đặt quân hậu tiếp theo như thế nào. Theo luật cờ vua, một quân hậu cớ thế ăn các quẩn khác nếu nằm trên cùng 1 đường, đường này có thể là :

- Hàng

- Cột

- Các đường chéo (đi qua tọa độ vị trí của hậu).

Suy ra rằng mồi hàng chỉ có thể chứa 1 và chỉ 1 quân hậu. Nên việc chọn vị trí cho quân hậu thứ i có thể giới hạn được ở hàng thứ i. Như thế tham số i trở thành chỉ hàng, và quá trình chọn vị trí cho quân hậu tiến hành trên toàn giá trị có thể có của các cột j.

\* quy ước:

x[i] // Chỉ quân hậu thứ i: nằm ở hàng L

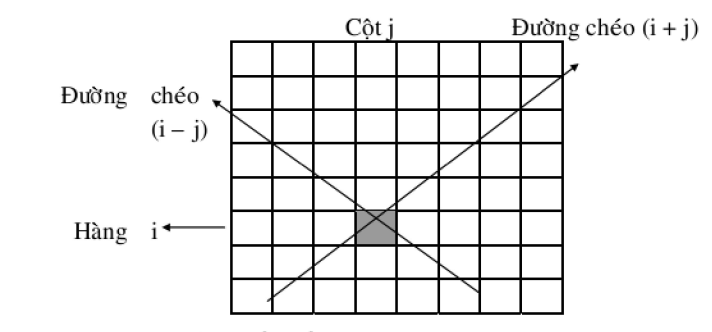
x[i] =j //quân hậu thứ i đặt ở cột j

Để quân hậu i (trên hàng i) chấp nhận cột j thì cột j và 2 đường chéo qua ô <i, j> phải còn trống tức là không có quân hậu khác chiếm lỉnh)

Lưu ý rằng trong 2 đường chéo :

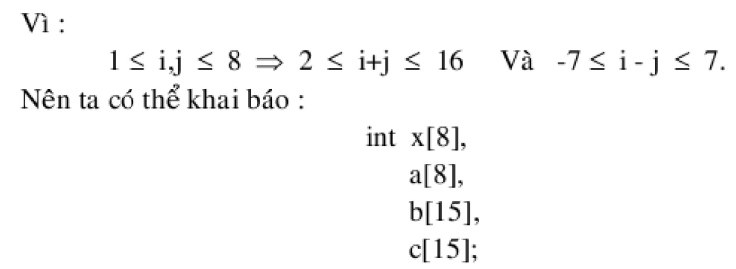
- Đường chéo ngược (vuông góc với đường chéo chính): tất cả các ồ đều có tổng 2 tọa độ i và j là hằng;

- Đường chéo thuận (song song với đường chéo chính) : gồm tất cả các ô (i,j) mà có hiệu các tọa độ (i-j) là hằng số.



Do đó ta sẽ chọn các mảng Boole 1 chiều để biểu diễn các trạng thái này : a[j] = 1 : Có nghĩa là không có quân hậu nào ở cột.

b[i+j 1=1: Có nghĩa là khỗng có quân hậu nào ở đường chéo ngược (i+j). c[i -j] = 1 : Cố nghĩa là không có quân hậu nào à đường chéo thuận (i-j).



Với các dữ liệu đã cho, thì lệnh đặt quân hậu sẽ thể hiện bởi :

x[ i ] = j; // đặt quân hậu thứ i trên cột j.

a[ j ] = 0;//Khi đặt hậu tại cột j , thì cột j không còn trống nữa

b[ i+ j ] = 0;//Các đường chéo tương ứng cùng không còn trống nữa .

c[ i - j ] = 0; //Các đường chéo tương ứng cùng không còn trống nữa .

\* Lệnh dời quân hậu là:

//Làm cho hàng i và các đường chéo tương ứng trở thành trống

a[j] = 1; b[i+j] = 1; c[i-j] = 1;

Còn điều kiện an toàn là ô có tọa độ (i, j ) nằm ờ hàng và các đường chéo chưa bị chiếm (được thể hiện bằng trị True). Do đó, có thể được thể hiện bởi biểu thức logic: **(a[j] && b[i+j] && c[i-j])**

**3. Thực hiện từng bước việc đặt 5, 6, 7 con hậu đầu tiên trên bàn cờ 8x8;**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Bước**  **Cách** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|
| **1** | **1** | **5** | **8** | **6** | **3** | **7** | **2** |
| **2** | **1** | **6** | **8** | **3** | **7** | **4** | **2** |
| **3** | **1** | **7** | **4** | **6** | **8** | **2** | **5** |
| **4** | **1** | **7** | **5** | **8** | **2** | **4** | **6** |
| **5** | **2** | **4** | **6** | **8** | **3** | **1** | **7** |
| **6** | **2** | **5** | **7** | **1** | **3** | **8** | **6** |
| **7** | **2** | **5** | **7** | **4** |  | **1** | **8** |
| **8** | **2** | **6** | **1** | **7** | **4** | **8** | **3** |
| **9** | **2** | **6** | **8** | **3** | **1** | **4** | **7** |
| **10** | **2** | **7** | **3** | **6** | **8** | **5** | **1** |
| **11** | **2** | **7** | **5** | **8** | **1** | **4** | **6** |
| **12** | **2** | **8** | **6** | **1** | **3** | **5** | **7** |
| **13** | **3** | **1** | **7** | **5** | **8** | **2** | **4** |
| **14** | **3** | **5** | **2** | **8** | **1** | **7** | **4** |
| **15** | **3** | **5** | **2** | **8** | **6** | **4** | **7** |
| **16** | **3** | **5** | **7** | **1** | **4** | **2** | **8** |
| **17** | **3** | **5** | **8** | **4** | **1** | **7** | **2** |
| **18** | **3** | **6** | **2** | **5** | **8** | **1** | **7** |
| **19** | **3** | **6** | **2** | **7** | **1** | **4** | **8** |
| **20** | **3** | **6** | **2** | **7** | **5** | **1** | **8** |
| **21** | **3** | **6** | **4** | **1** | **8** | **5** | **7** |
| **22** | **3** | **6** | **4** | **2** | **8** | **5** | **7** |
| **23** | **3** | **6** | **8** | **1** | **4** | **7** | **5** |
| **24** | **3** | **6** | **8** | **1** | **5** | **7** | **2** |
| **25** | **3** | **6** | **8** | **2** | **4** | **1** | **7** |
| **26** | **3** | **7** | **2** | **8** | **5** | **1** | **4** |
| **27** | **3** | **7** | **2** | **8** | **6** | **4** | **1** |
| **28** | **3** | **8** | **4** | **7** | **1** | **6** | **2** |
| **29** | **4** | **1** | **5** | **8** | **2** | **7** | **3** |
| **30** | **4** | **1** | **5** | **8** | **6** | **3** | **7** |
| **31** | **4** | **2** | **6** | **8** | **6** | **1** | **3** |
| **32** | **4** | **2** | **7** | **3** | **6** | **8** | **1** |
| **33** | **4** | **2** | **7** | **3** | **6** | **8** | **5** |
| **34** | **4** | **2** | **7** | **5** | **1** | **8** | **6** |
| **35** | **4** | **2** | **8** | **5** | **7** | **1** | **3** |
| **36** | **4** | **2** | **8** | **6** | **1** | **3** | **5** |
| **37** | **4** | **6** | **1** | **5** | **2** | **8** | **3** |
| **38** | **4** | **6** | **8** | **2** | **7** | **1** | **3** |
| **39** | **4** | **6** | **8** | **3** | **1** | **7** | **5** |
| **40** | **4** | **7** | **1** | **8** | **5** | **2** | **6** |
| **41** | **4** | **7** | **3** | **8** | **2** | **5** | **1** |
| **42** | **4** | **7** | **5** | **2** | **6** | **1** | **3** |
| **43** | **4** | **7** | **5** | **3** | **1** | **6** | **8** |
| **44** | **4** | **8** | **1** | **3** | **6** | **2** | **7** |
| **45** | **4** | **8** | **1** | **5** | **7** | **2** | **6** |
| **46** | **4** | **8** | **5** | **3** | **1** | **7** | **2** |
| **47** | **5** | **1** | **4** | **6** | **8** | **2** | **7** |
| **48** | **5** | **1** | **8** | **4** | **2** | **7** | **3** |
| **49** | **5** | **1** | **8** | **6** | **3** | **7** | **2** |
| **50** | **5** | **2** | **4** | **6** | **8** | **3** | **1** |
| **51** | **5** | **2** | **4** | **7** | **3** | **8** | **6** |
| **52** | **5** | **2** | **6** | **1** | **7** | **4** | **8** |
| **53** | **5** | **2** | **8** | **1** | **4** | **7** | **3** |
| **54** | **5** | **3** | **1** | **6** | **8** | **2** | **4** |
| **55** | **5** | **3** | **1** | **7** | **2** | **8** | **6** |
| **56** | **5** | **3** | **8** | **4** | **7** | **1** | **6** |
| **57** | **5** | **7** | **1** | **3** | **8** | **6** | **4** |
| **58** | **5** | **7** | **1** | **4** | **2** | **8** | **6** |
| **59** | **5** | **7** | **2** | **4** | **8** | **1** | **3** |
| **60** | **5** | **7** | **2** | **6** | **3** | **1** | **4** |
| **61** | **5** | **7** | **2** | **6** | **3** | **1** | **8** |
| **62** | **5** | **7** | **4** | **1** | **3** | **8** | **6** |
| **63** | **5** | **8** | **4** | **1** | **3** | **6** | **2** |
| **64** | **5** | **8** | **4** | **1** | **7** | **2** | **6** |
| **65** | **6** | **1** | **5** | **2** | **8** | **3** | **7** |
| **66** | **6** | **2** | **7** | **1** | **3** | **5** | **8** |
| **67** | **6** | **2** | **7** | **1** | **4** | **8** | **5** |
| **68** | **6** | **3** | **1** | **7** | **5** | **8** | **2** |
| **69** | **6** | **3** | **1** | **8** | **4** | **2** | **7** |
| **70** | **6** | **3** | **1** | **8** | **5** | **2** | **4** |
| **71** | **6** | **3** | **5** | **7** | **1** | **4** | **2** |
| **72** | **6** | **3** | **5** | **8** | **1** | **4** | **2** |
| **73** | **6** | **3** | **7** | **2** | **4** | **8** | **1** |
| **74** | **6** | **3** | **7** | **2** | **8** | **5** | **1** |
| **75** | **6** | **3** | **7** | **4** | **1** | **8** | **2** |
| **76** | **6** | **4** | **1** | **5** | **8** | **2** | **7** |
| **77** | **6** | **4** | **2** | **8** | **5** | **7** | **1** |
| **78** | **6** | **4** | **7** | **1** | **3** | **5** | **2** |
| **79** | **6** | **4** | **7** | **1** | **8** | **2** | **5** |
| **80** | **6** | **8** | **2** | **4** | **1** | **7** | **5** |
| **81** | **7** | **1** | **3** | **8** | **6** | **4** | **2** |
| **82** | **7** | **2** | **4** | **1** | **8** | **5** | **3** |
| **83** | **7** | **2** | **6** | **3** | **1** | **4** | **8** |
| **84** | **7** | **3** | **1** | **6** | **8** | **5** | **2** |
| **85** | **7** | **3** | **8** | **2** | **5** | **1** | **6** |
| **86** | **7** | **4** | **2** | **5** | **8** | **1** | **3** |
| **87** | **7** | **4** | **2** | **8** | **6** | **1** | **3** |
| **88** | **7** | **5** | **3** | **1** | **6** | **8** | **2** |
| **89** | **8** | **2** | **4** | **1** | **7** | **5** | **3** |
| **90** | **8** | **2** | **5** | **3** | **1** | **7** | **4** |
| **91** | **8** | **3** | **1** | **6** | **2** | **5** | **7** |
| **92** | **8** | **4** | **1** | **3** | **6** | **2** | **7** |

**4. Viết chương trình sử dụng C, C++;**

void thu(int i)

{

int j;

if (i<=7)

{

for (j=0; j<8; j++)

{

if (a[j] == 1 && b[i+j] ==1 && c[i-j+7] == 1)//chua co quan hau nao o cot j

{

x[i] = j;

a[j] = 0; b[i+j] = 0; c[i-j+7] = 0;

thu(i+1);

a[j] = 1; b[i+j] = 1; c[i-j+7] = 1;

}

}

}

else

{

print();dem++;

}

}

void TamHau()

{

int i;

for (i=0; i<8; i++)

{

a[i] = 1;

x[i] = -1;

}

for (i=0; i<15; i++)

{

b[i] = 1;

c[i] = 1;

}

thu(0);

printf("\n Tong so cach dat quan Hau: %d",dem);

}

**5. Đánh giá độ phức tạp thuật toán theo lý thuyết: O(n!)**