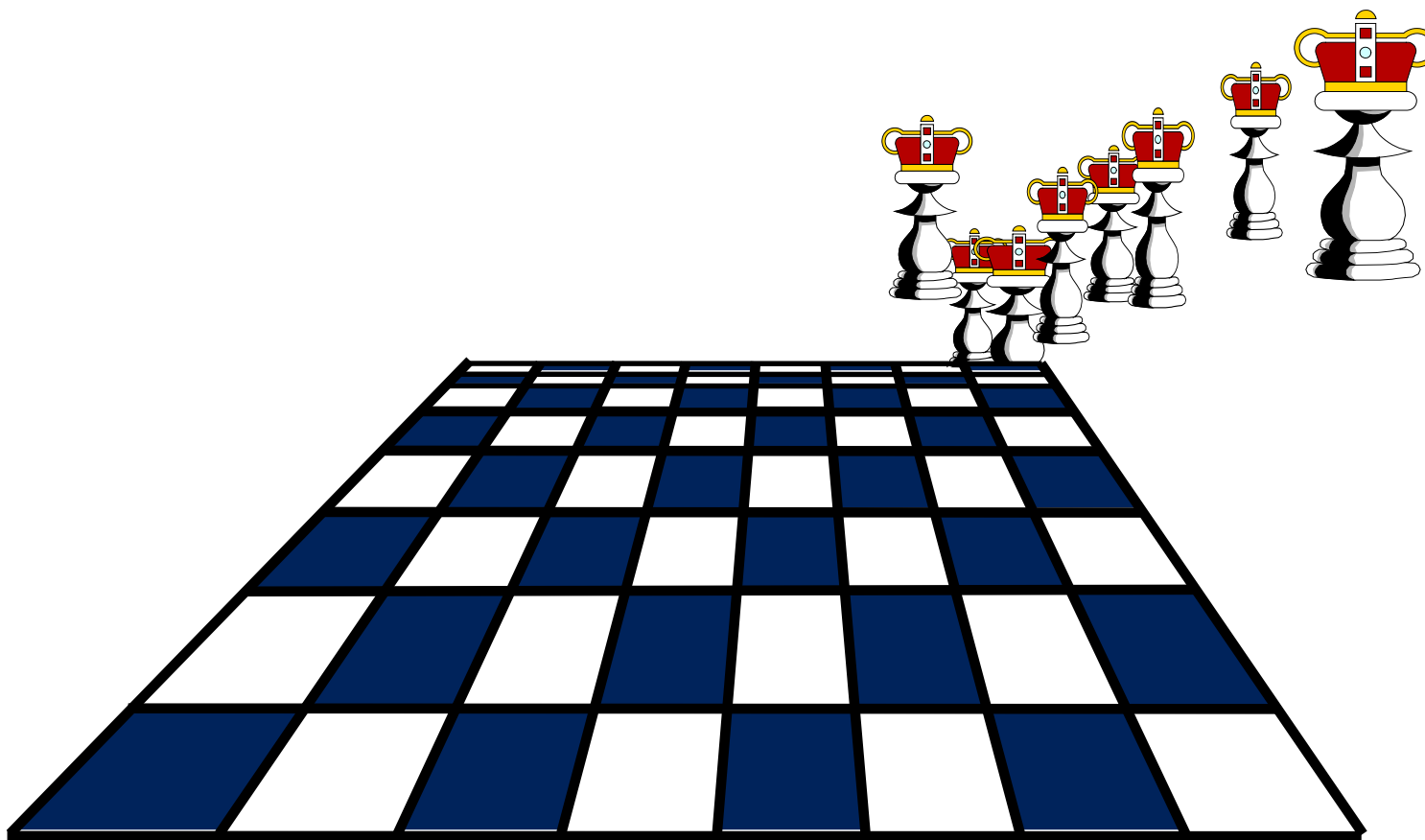
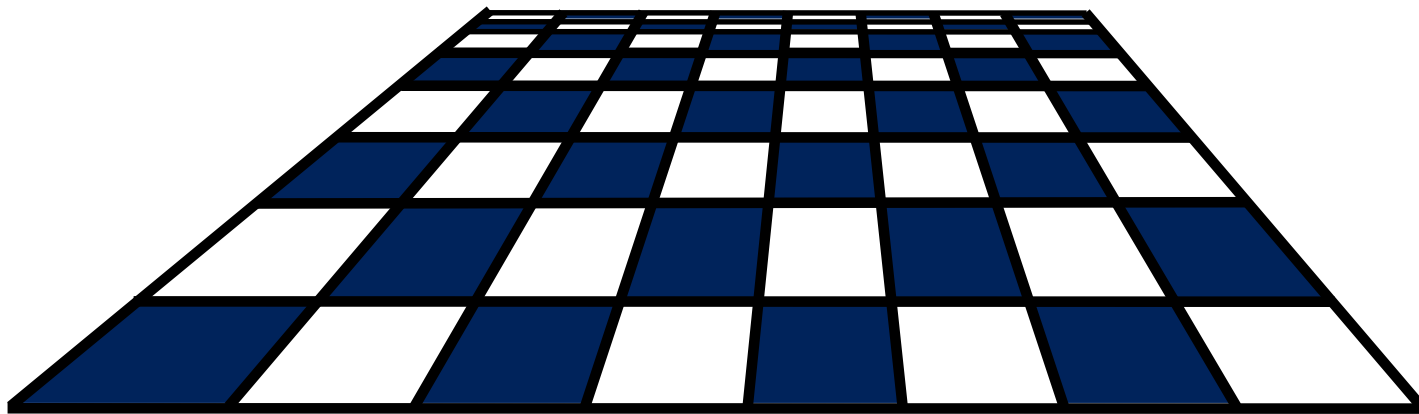
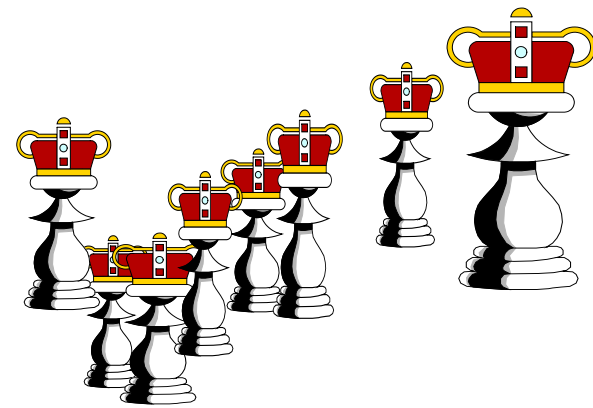


The n -Queens Problem



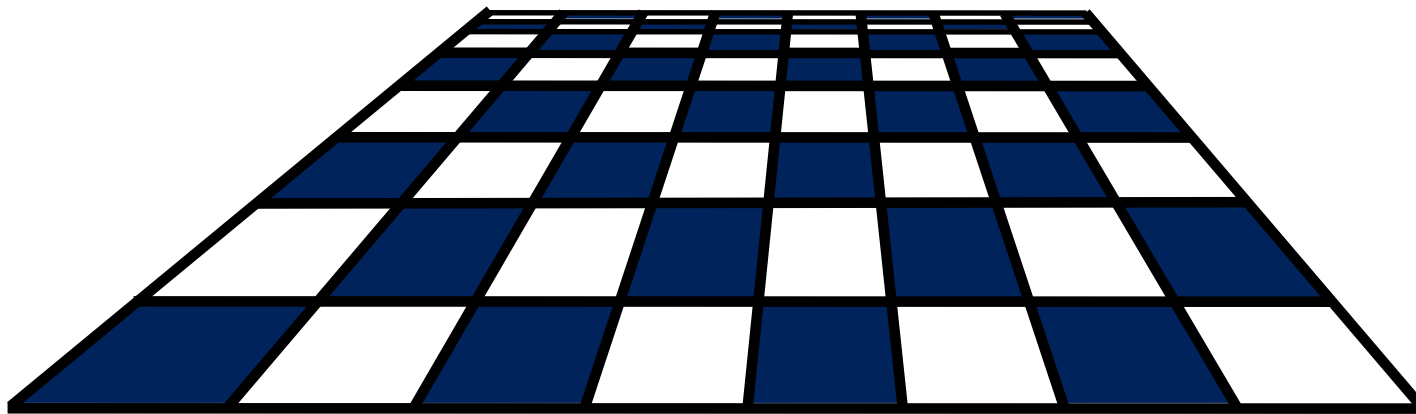
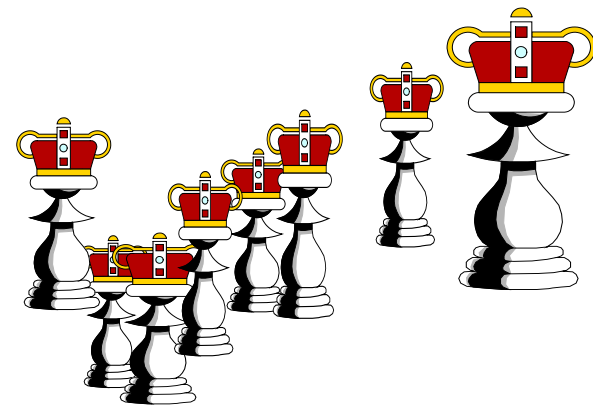
The n -Queens Problem

- Giả sử ta có 8 con hậu...
- ...và bàn cờ quốc tế



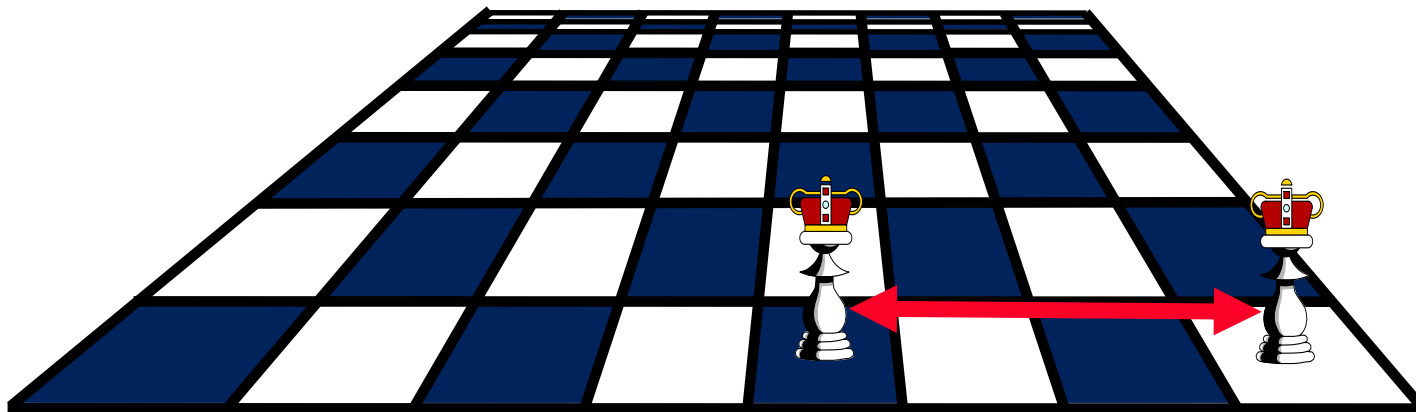
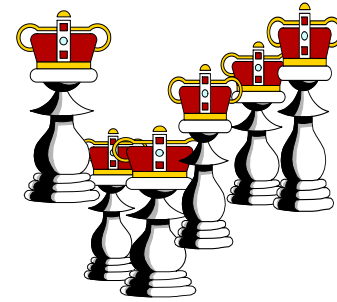
The n -Queens Problem

*Có thể xếp các con hậu
sao cho không có hai
con nào ăn nhau hay
không ?*



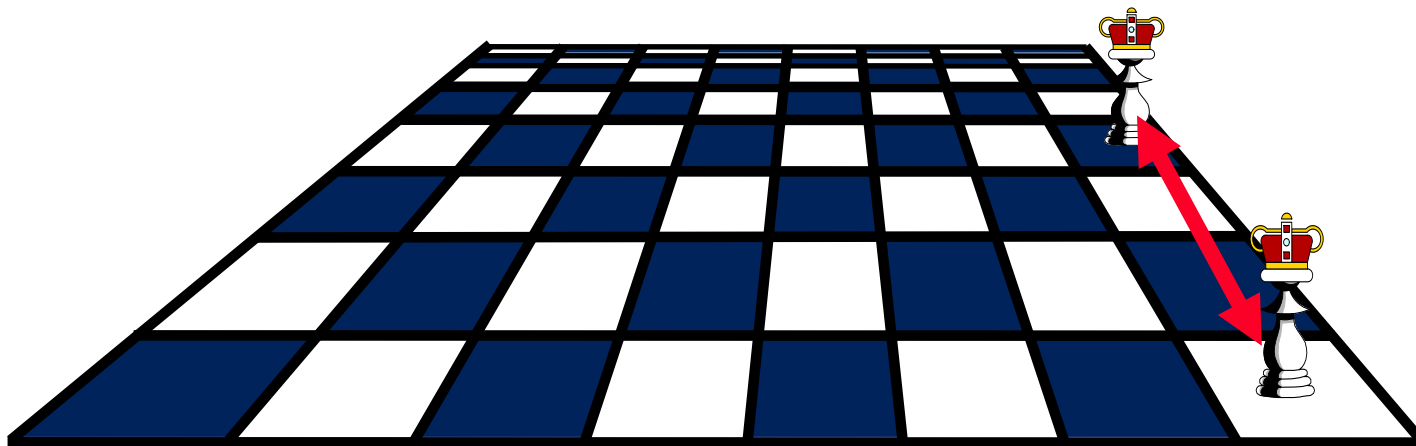
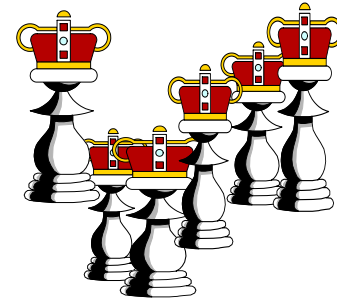
The n-Queens Problem

Hai con hậu bất kỳ
không được xếp trên
cùng một dòng ...



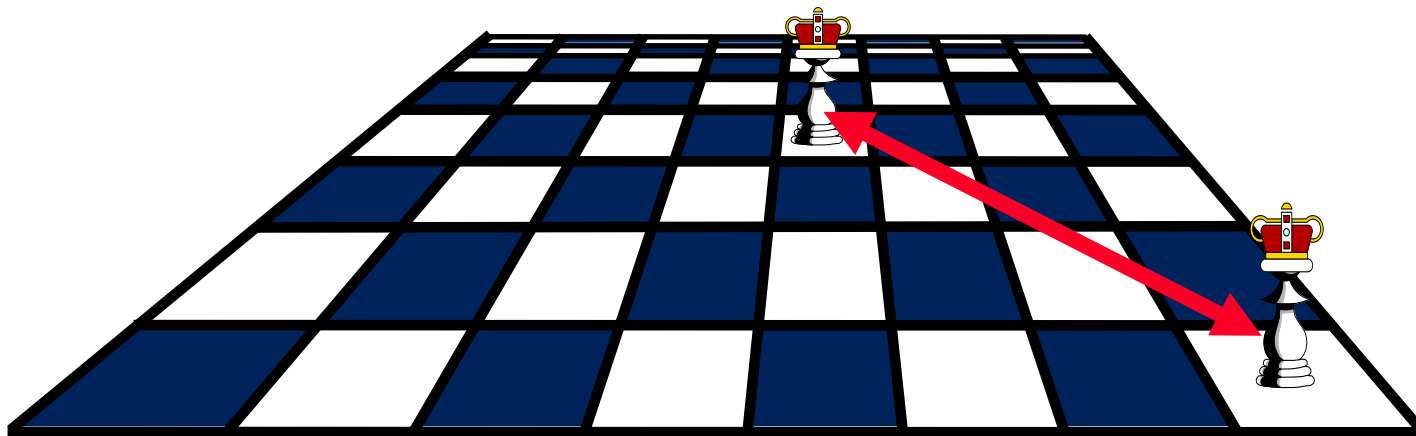
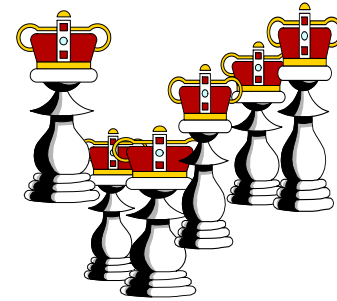
The n-Queens Problem

Hai con hậu bất kỳ
không được xếp trên
cùng một cột ...



The n-Queens Problem

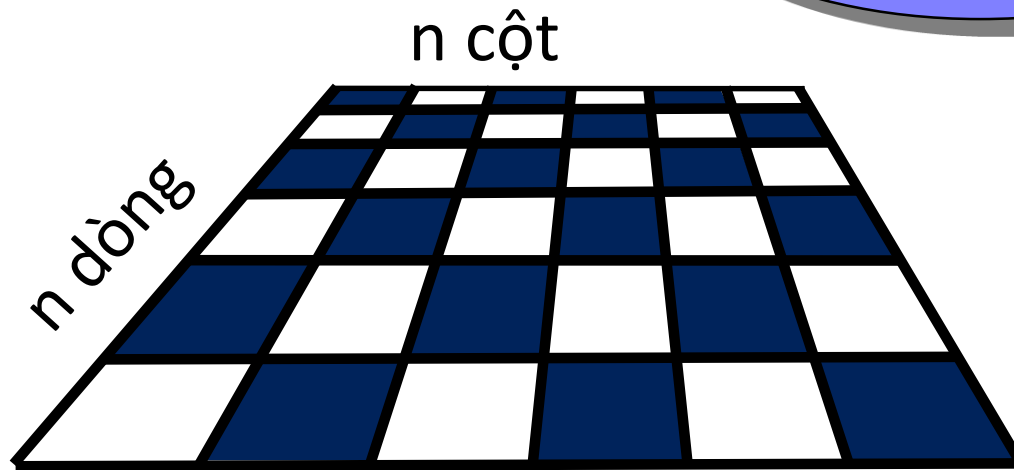
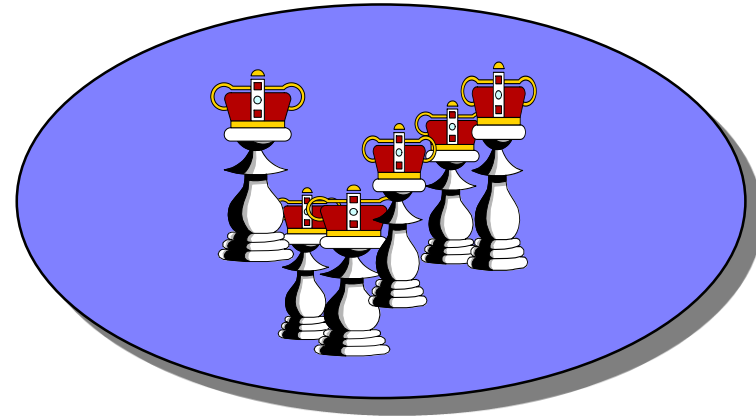
Hai con hậu bất kỳ
không được xếp trên
cùng một đường chéo!



The n-Queens Problem

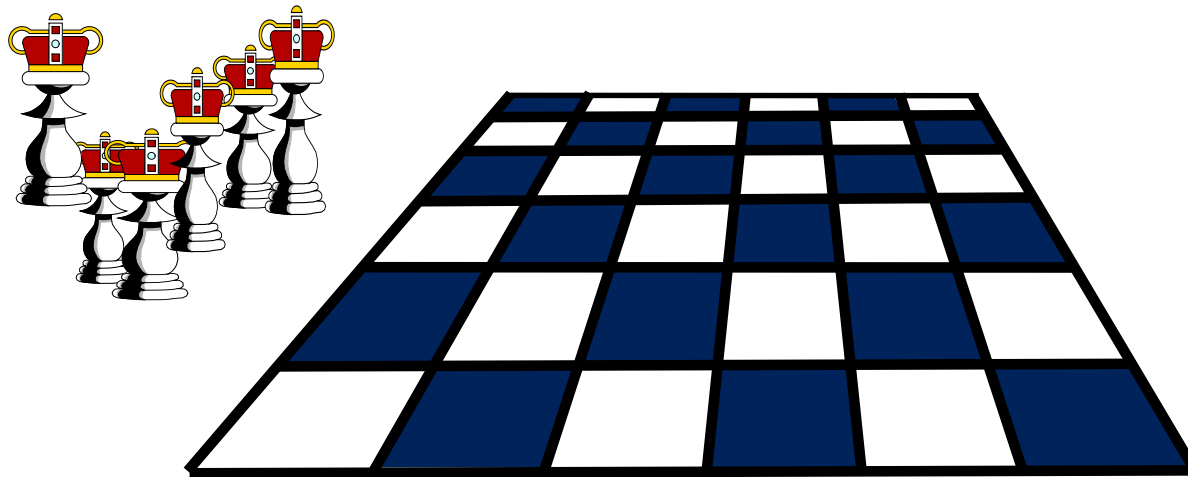
Kích thước $n \times n$

n con hậu



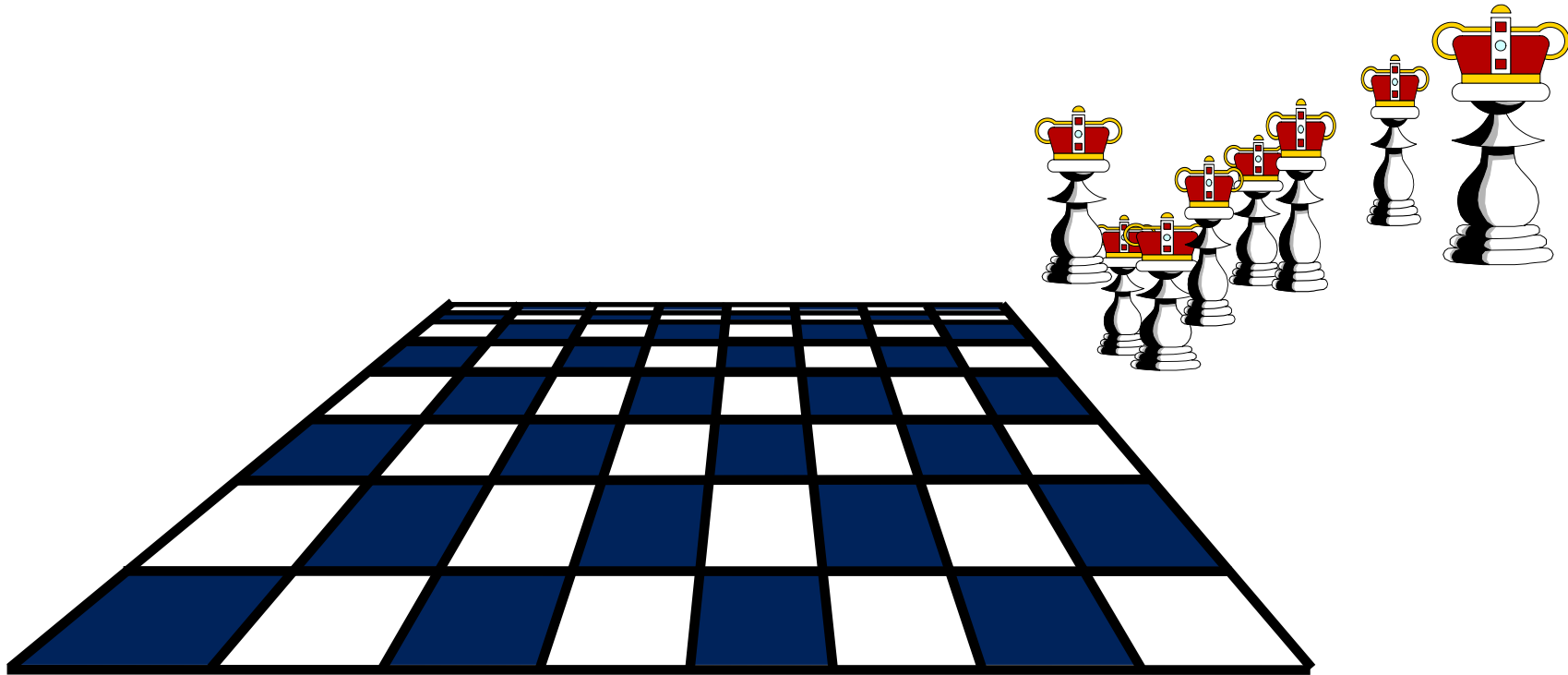
The n-Queens Problem

Xét bài toán xếp n con
hậu lên bàn cờ kích
thước $n \times n$.



Bài toán xếp hậu

- Liệt kê tất cả các cách xếp n quân Hậu trên bàn cờ $n \times n$ sao cho chúng không ăn được lẫn nhau, nghĩa là sao cho không có hai con nào trong số chúng nằm trên cùng một dòng hay một cột hay một đường chéo của bàn cờ.



Biểu diễn lời giải

- Đánh số các cột và dòng của bàn cờ từ 1 đến n . Một cách xếp hậu có thể biểu diễn bởi bộ có n thành phần (a_1, a_2, \dots, a_n) , trong đó a_i là tọa độ cột của con Hậu ở dòng i .
- Các điều kiện đặt ra đối với bộ (a_1, a_2, \dots, a_n) :
 - $a_i \neq a_j$, với mọi $i \neq j$ (nghĩa là hai con hậu ở hai dòng i và j không được nằm trên cùng một cột);
 - $|a_i - a_j| \neq |i - j|$, với mọi $i \neq j$ (nghĩa là hai con hậu ở hai ô (a_i, i) và (a_j, j) không được nằm trên cùng một đường chéo).

Phát biểu bài toán

- Như vậy bài toán xếp Hậu dẫn về bài toán liệt kê các phần tử của tập:

$$D = \{ (a_1, a_2, \dots, a_n) \in N^n :$$

$$a_i \neq a_j \text{ và } |a_i - a_j| \neq |i - j|, i \neq j \}.$$

Hàm nhận biết ứng cử viên

```
int UCVh(int j, int k) {  
    // UCVh nhận giá trị 1  
    // khi và chỉ khi  $j \in S_k$   
    int i;  
    for (i=1; i<k; i++)  
        if ((j == a[i]) || (fabs(j-a[i]) == k-i)) return 0;  
    return 1;  
}
```

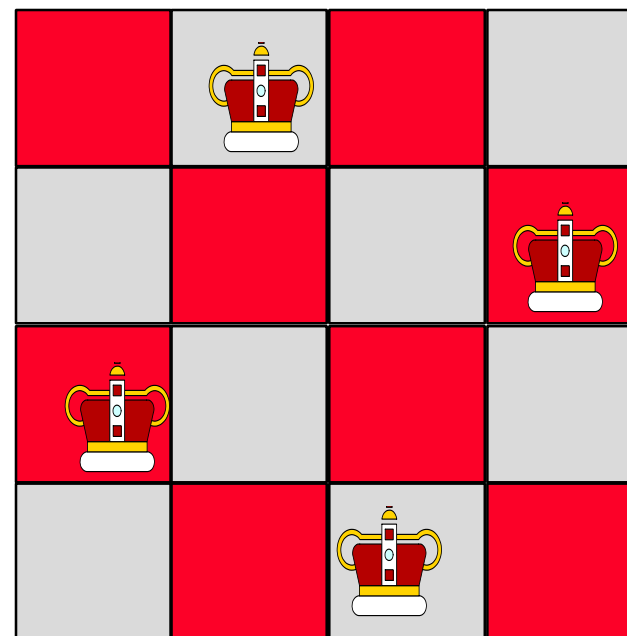
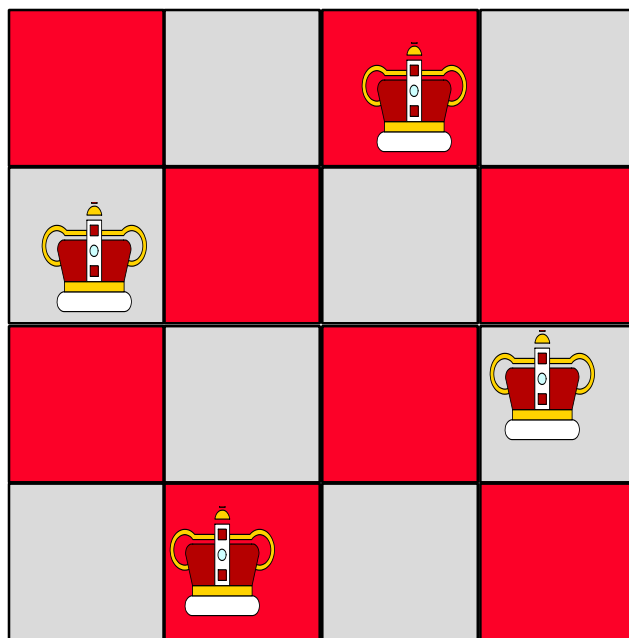
Chương trình trên C

```
#include <stdio.h>
#include <math.h>
int  n, count;
int  a[20];
int Ghinhan() {
    int i;
    count++; printf("%i. ",count);
    for (i=1; i<=n;i++) printf("%i  ",a[i]);
    printf("\n");
}

int UCVh(int j, int k) {
    /* UCVh nhận giá trị 1 khi và chỉ khi  $j \in S_k$  */
    int i;
    for (i=1; i<k; i++)
        if ((j==a[i])||(fabs(j-a[i])==k-i)) return 0;
    return 1;
}
```

```
int Hau(int i){    int j;
    for (j=1; j<=n; j++)
        if (UCVh(j, i)){
            a[i] = j;
            if (i == n) Ghinhan();
            else Hau(i+1);
        }
}

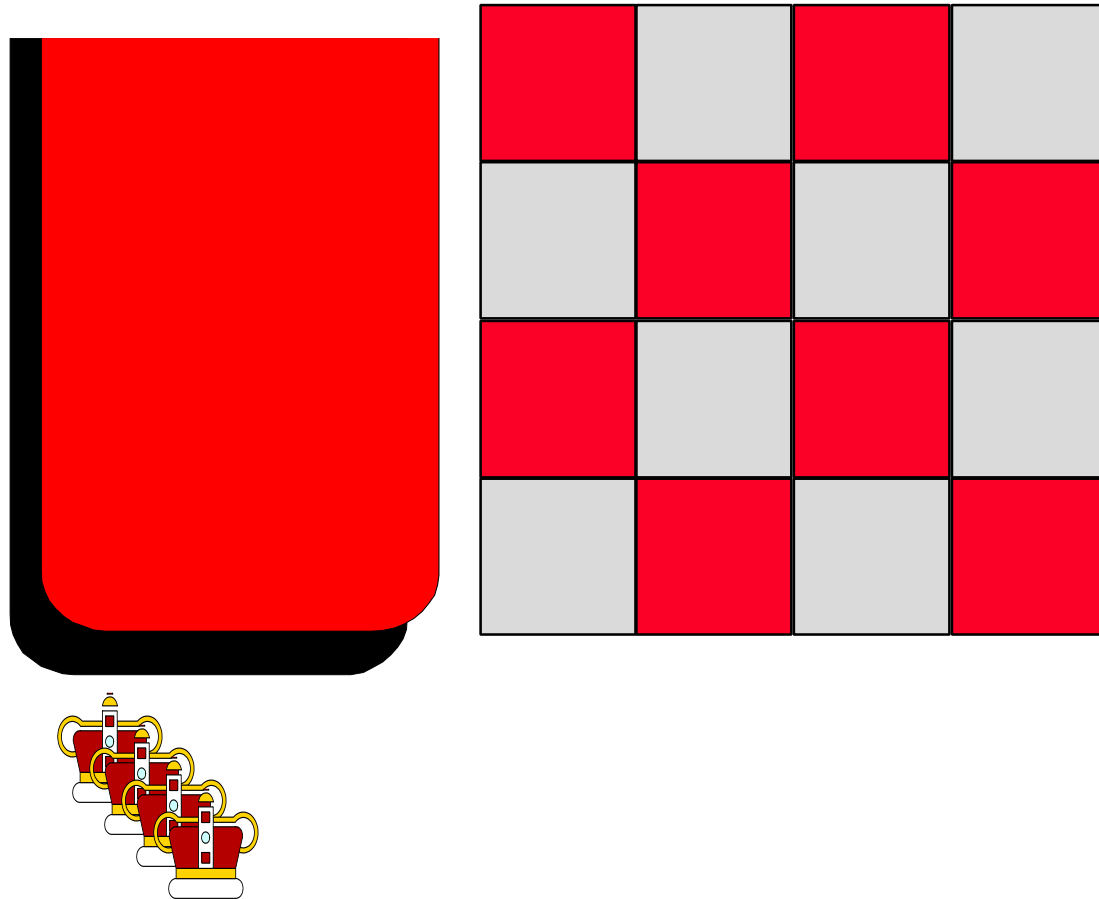
int main() {
    printf("Input n = "); scanf("%i",&n);
    printf("\n==== RESULT ====\n");
    count = 0; Hau(1);
    if (count == 0) printf("No solution!\n");
    getch();
    printf("\n Press Enter to finish... ");
    getchar();
}
```



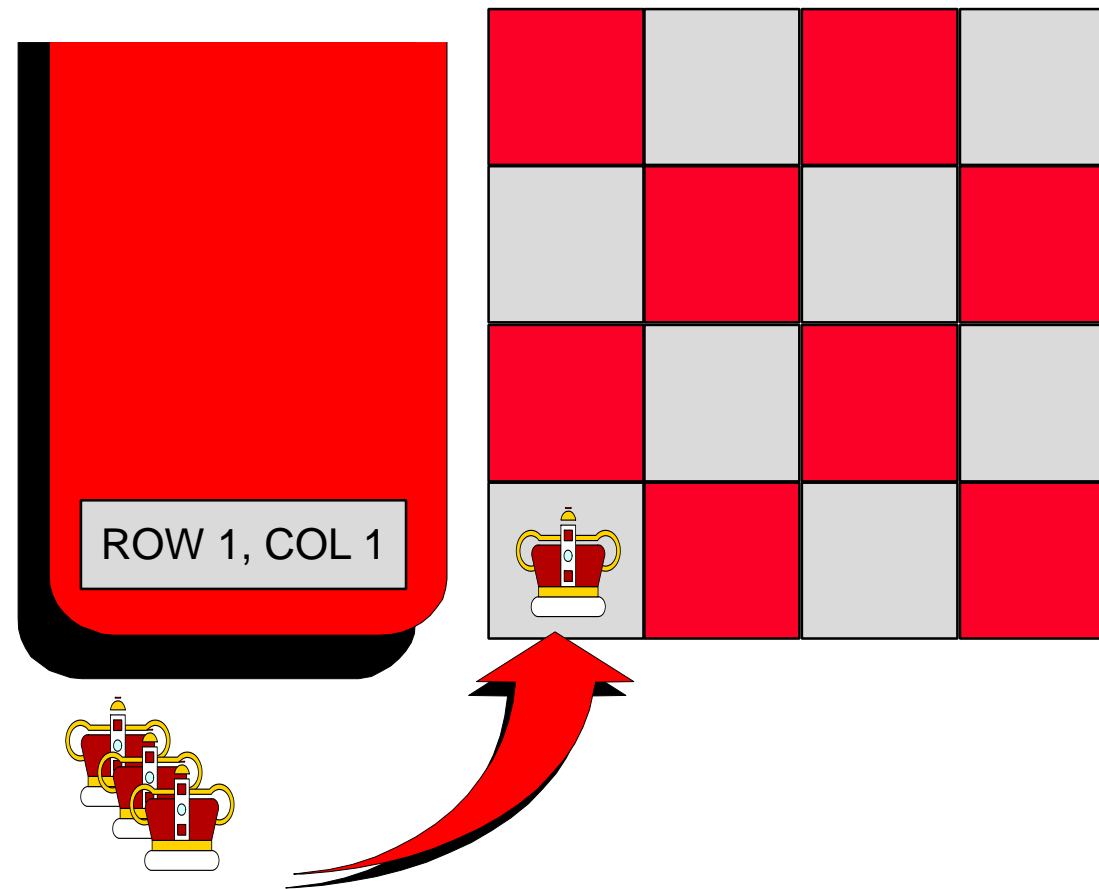
Chú ý

- Rõ ràng là bài toán xếp hậu không phải là luôn có lời giải, chẳng hạn bài toán không có lời giải khi $n=2, 3$. Do đó điều này cần được thông báo khi kết thúc thuật toán.
- Thuật toán trình bày ở trên là chưa hiệu quả. Nguyên nhân là ta đã không xác định được chính xác các tập UCV vào các vị trí của lời giải.

Thuật toán làm việc như thế nào

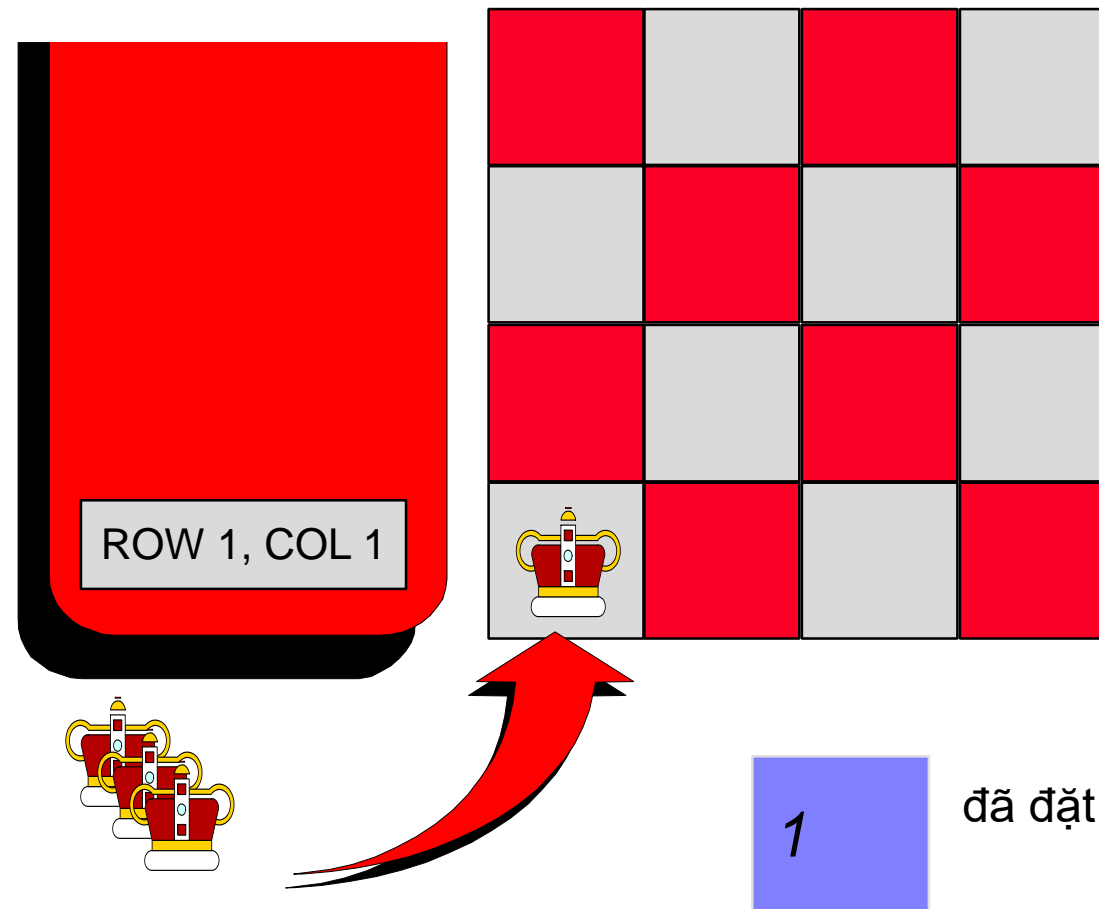


Thuật toán làm việc như thế nào



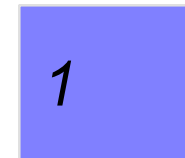
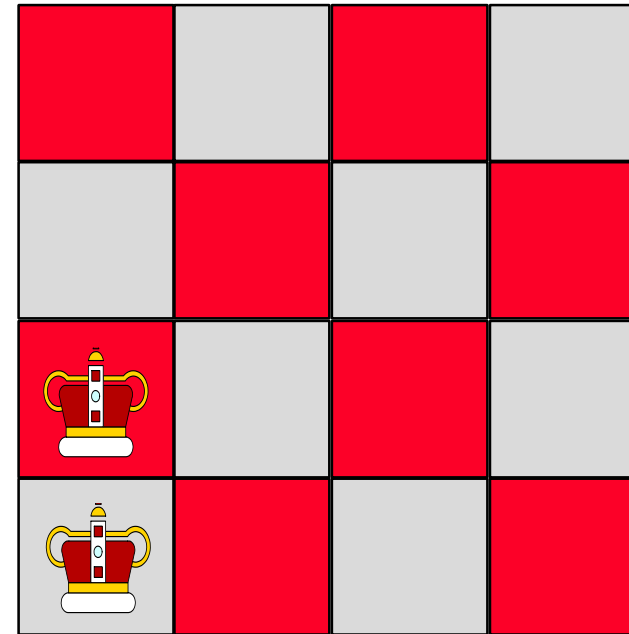
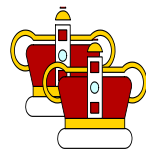
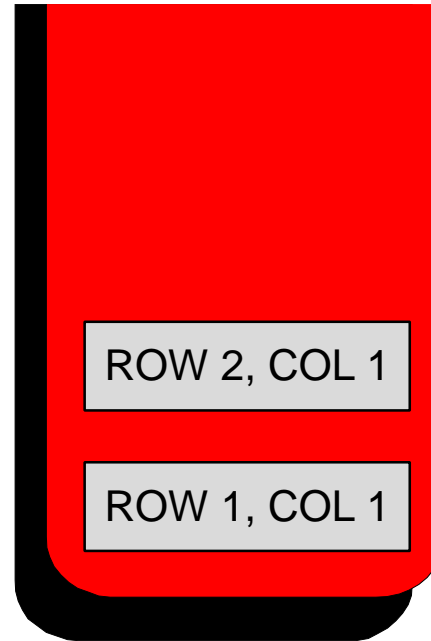
Thuật toán làm việc như thế nào

- Xếp con hậu ở dòng 1 vào vị trí cột 1



Thuật toán làm việc như thế nào

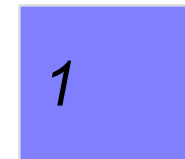
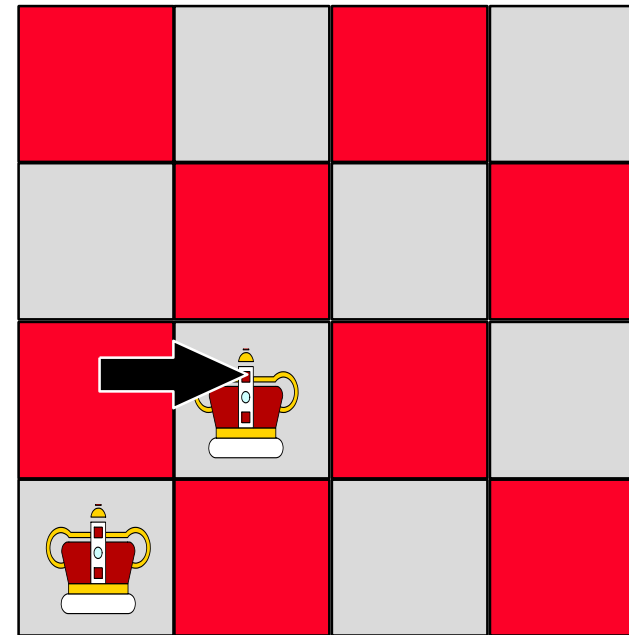
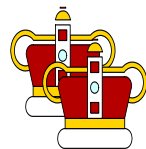
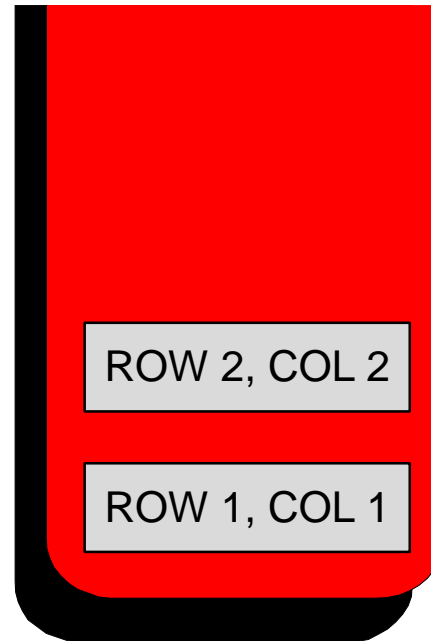
- Thử xếp con hậu ở dòng 2 vào vị trí cột 1



đã đặt

Thuật toán làm việc như thế nào

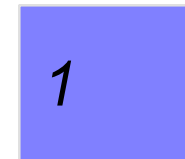
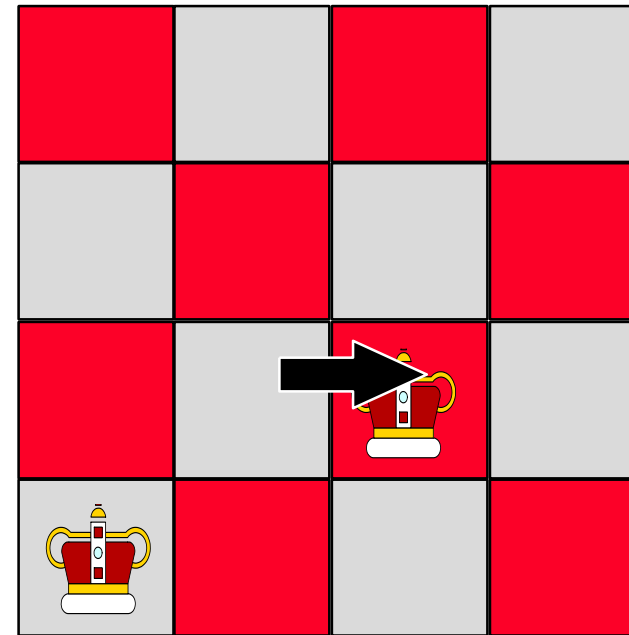
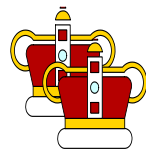
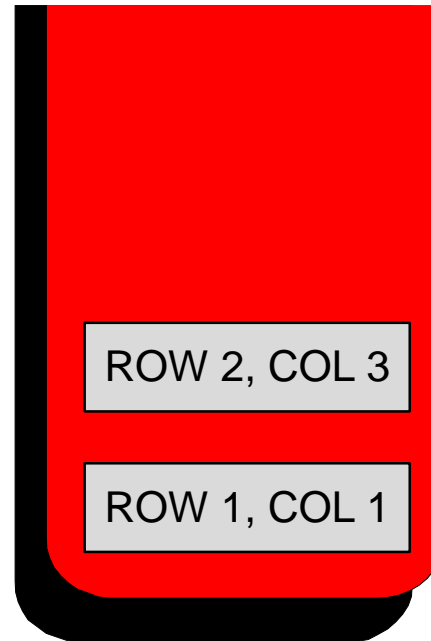
- Thử xếp con hậu ở dòng 2 vào vị trí cột 2



đã đặt

Thuật toán làm việc như thế nào

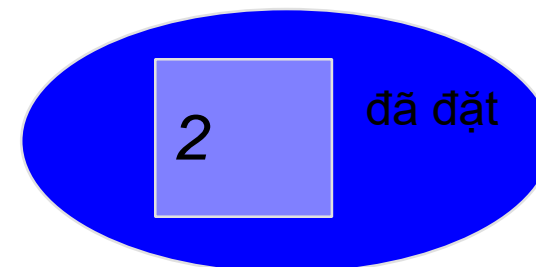
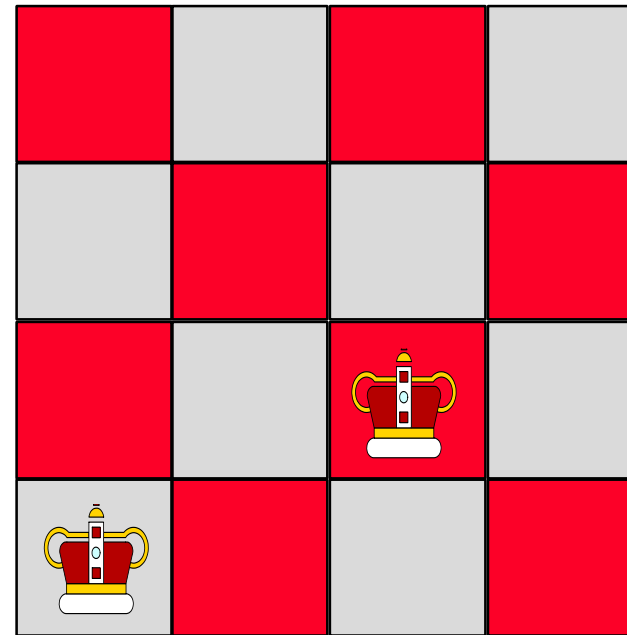
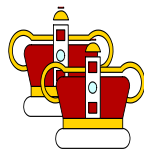
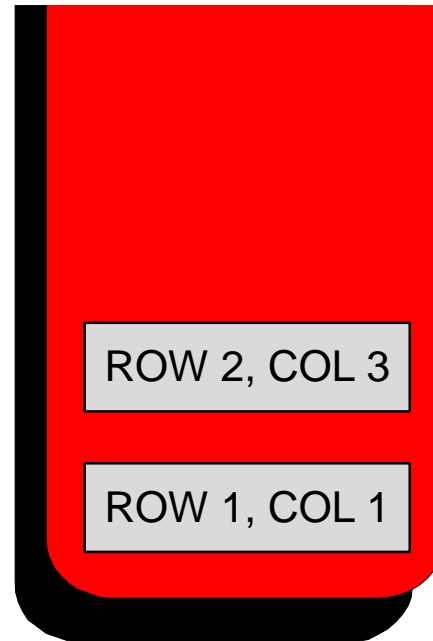
- Thử xếp con hậu ở dòng 2 vào vị trí cột 3



1 đã đặt

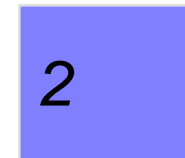
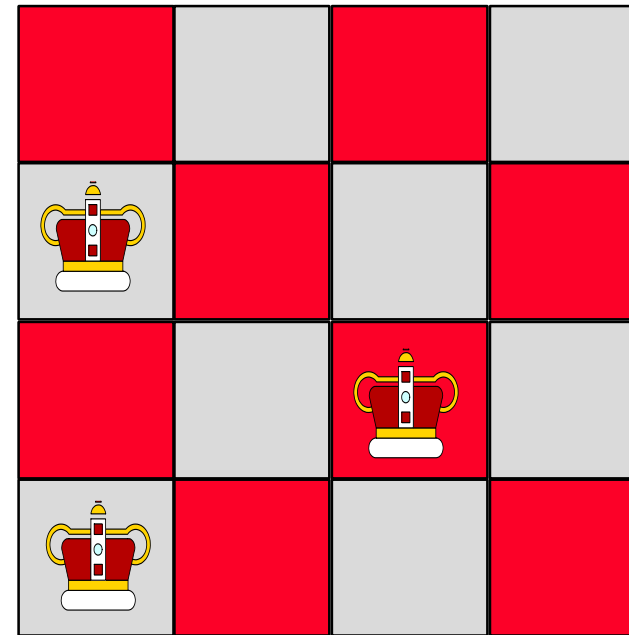
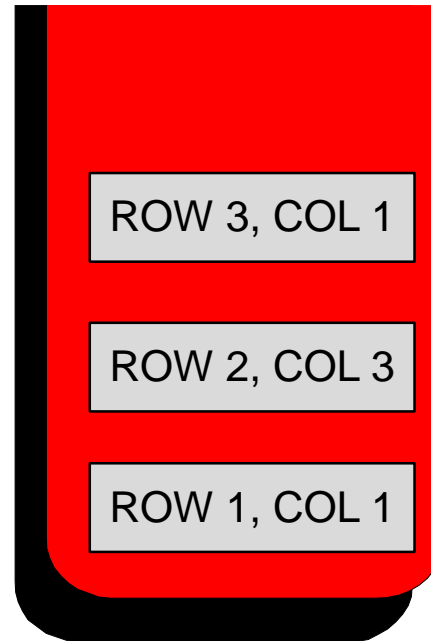
Thuật toán làm việc như thế nào

- Chấp nhận xếp con hậu ở dòng 2 vào vị trí cột 3



Thuật toán làm việc như thế nào

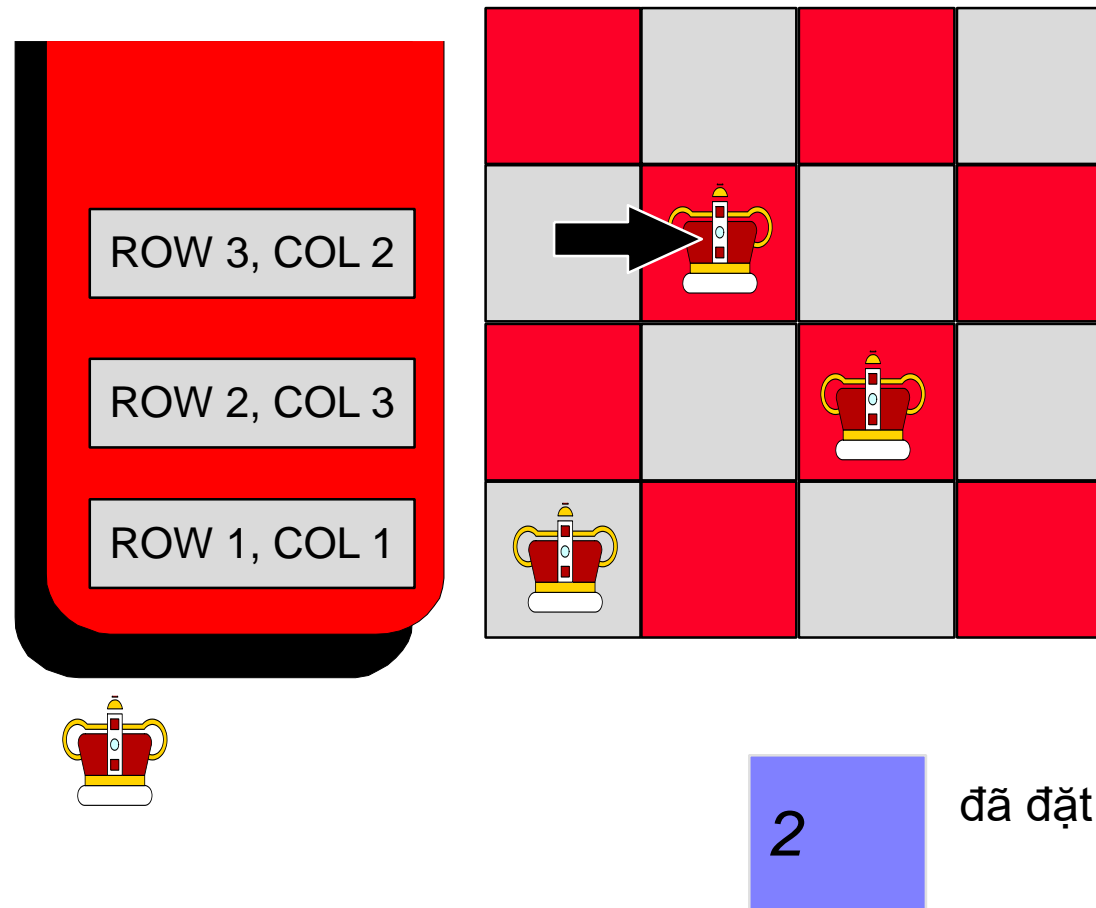
Thử xếp con hậu ở
dòng 3
vào cột đầu tiên



đã đặt

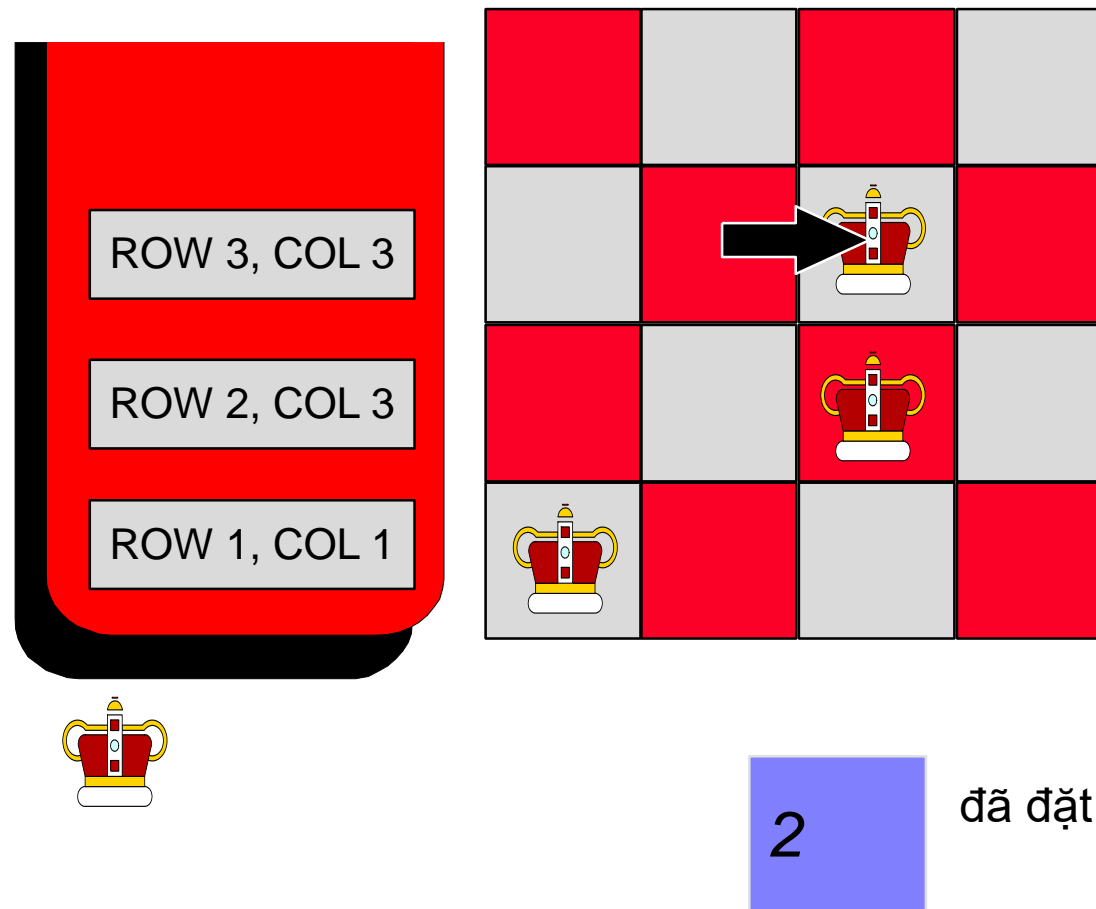
Thuật toán làm việc như thế nào

Thử cột tiếp theo



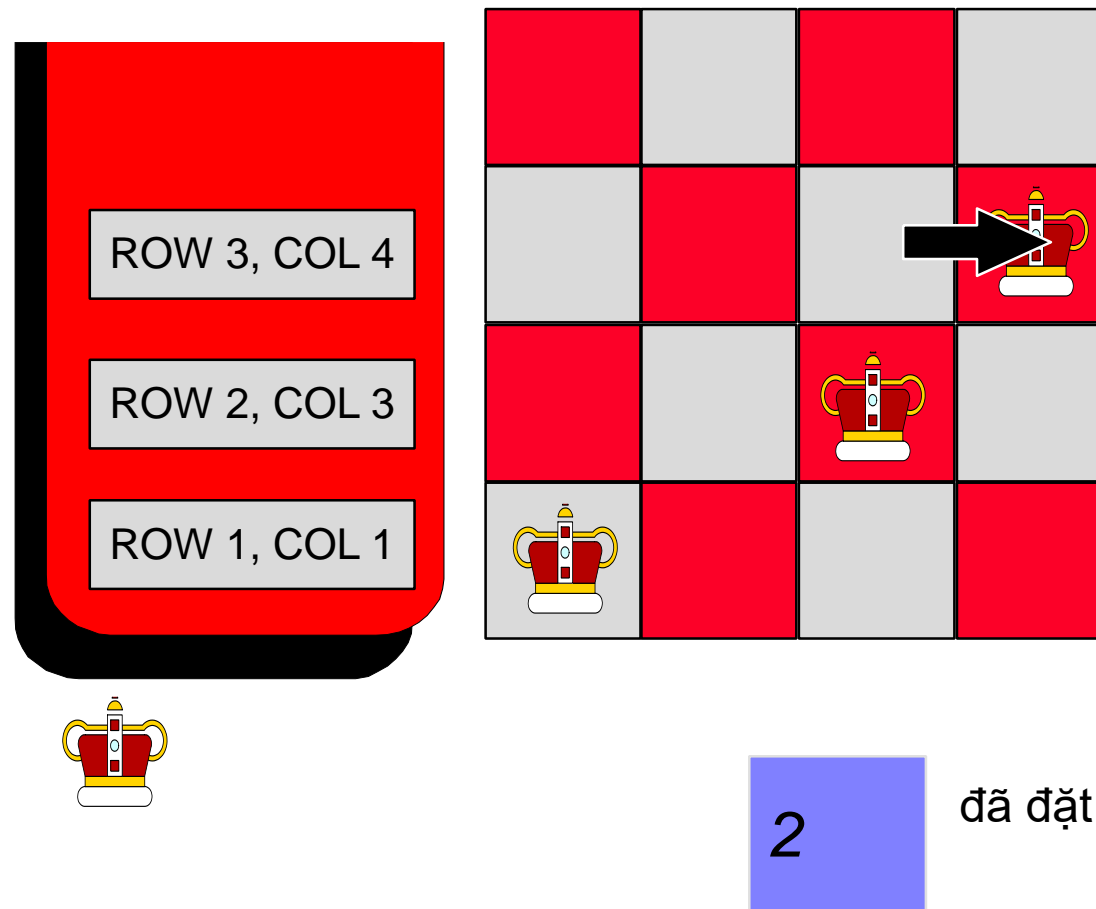
Thuật toán làm việc như thế nào

- Thử cột tiếp theo



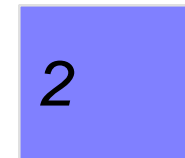
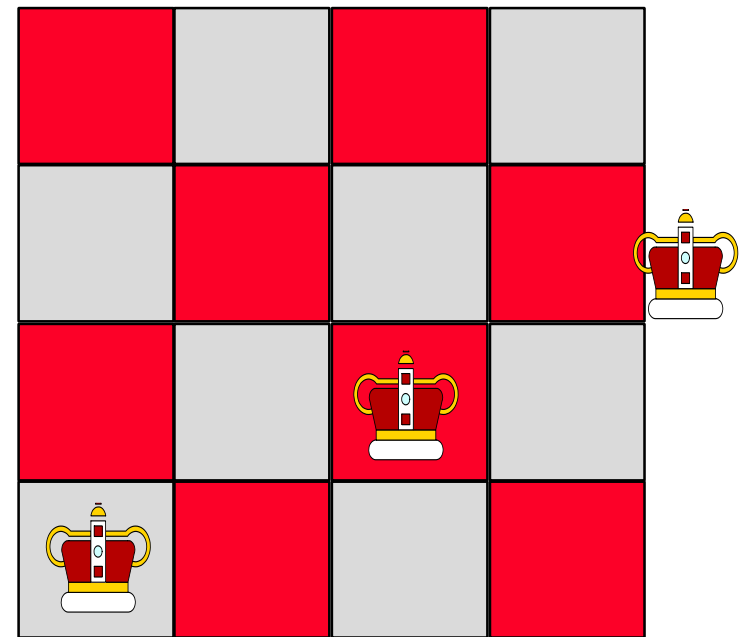
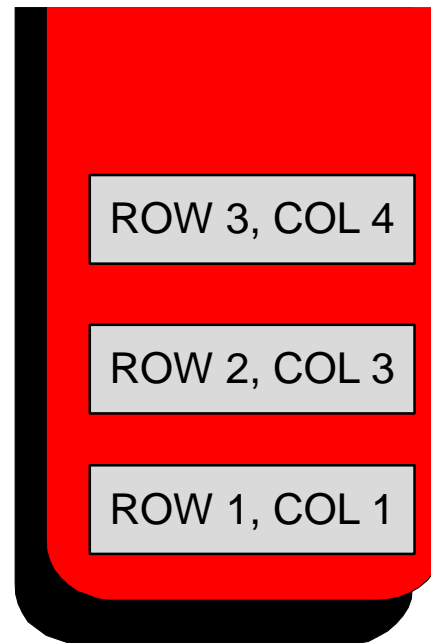
Thuật toán làm việc như thế nào

- Thử cột tiếp theo



Thuật toán làm việc như thế nào

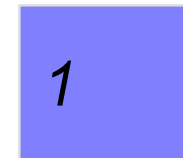
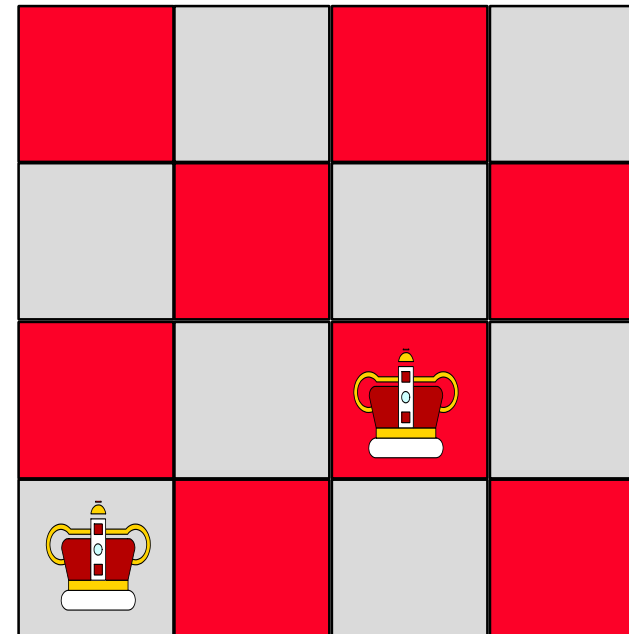
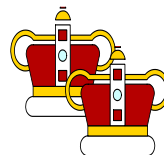
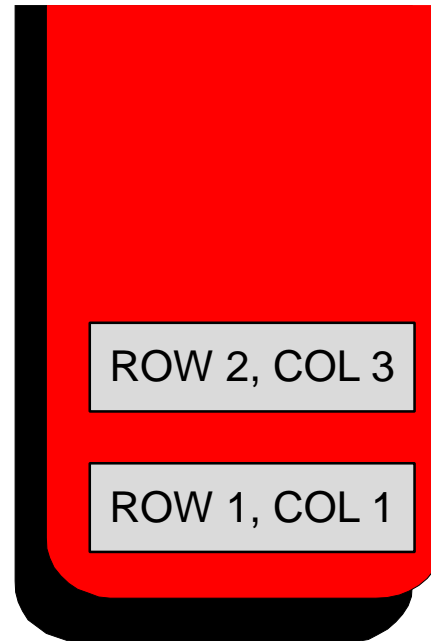
...không có vị trí
đặt con hậu ở
dòng 3.



đã đặt

Thuật toán làm việc như thế nào

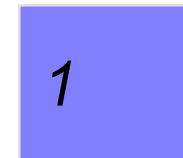
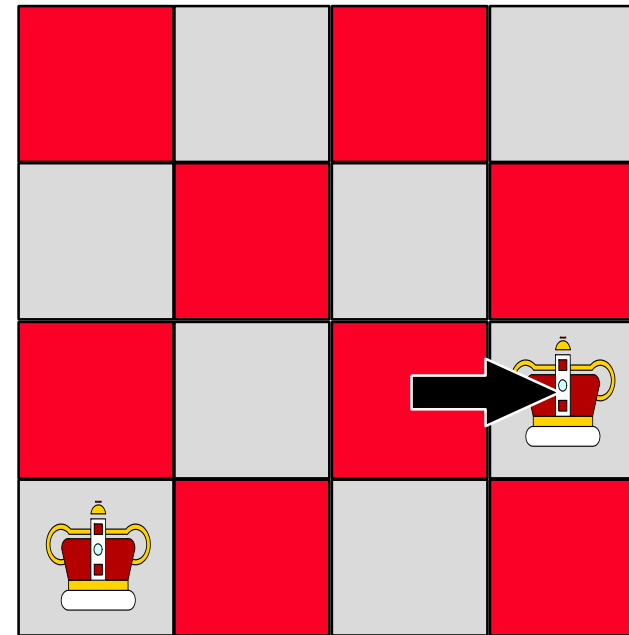
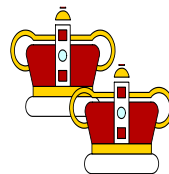
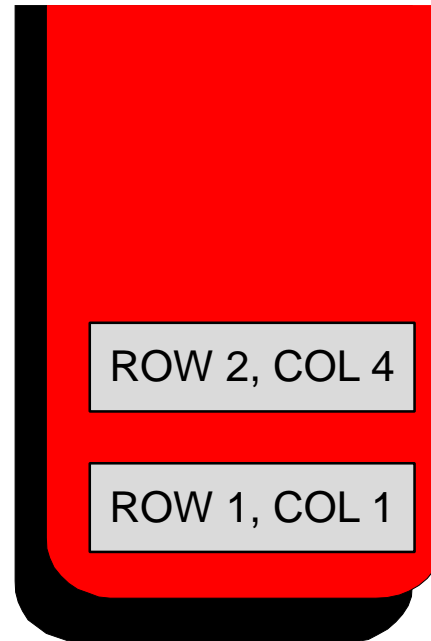
Quay lại dịch
chuyển con hậu ở
dòng 2



đã đặt

Thuật toán làm việc như thế nào

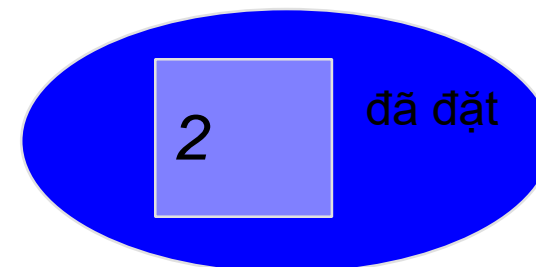
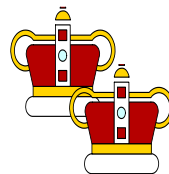
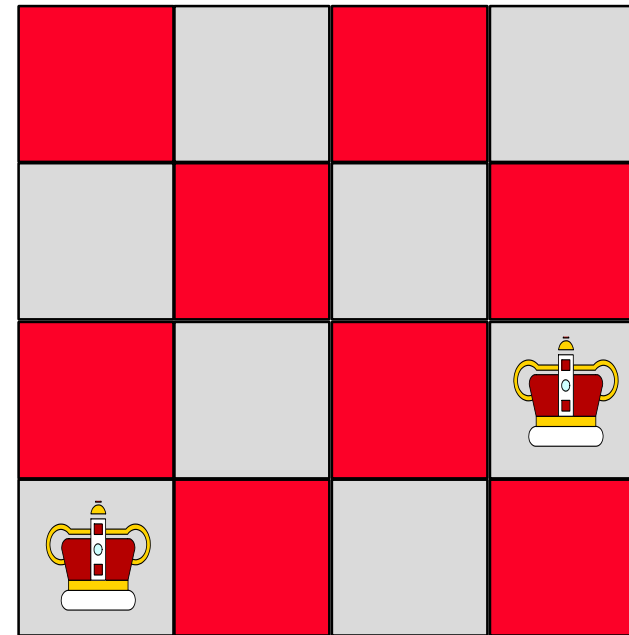
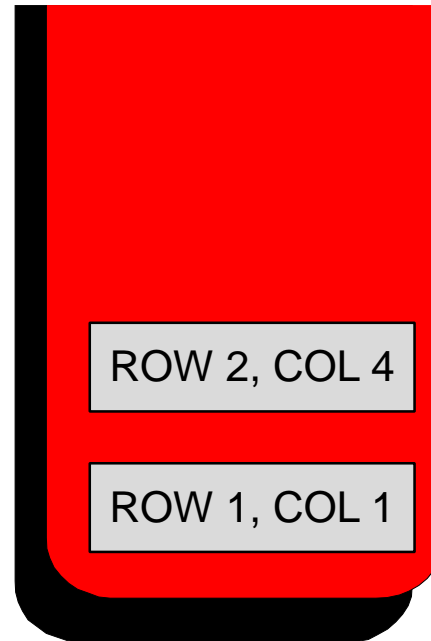
Đẩy con hậu ở
dòng 2 sang cột
thứ 4.



đã đặt

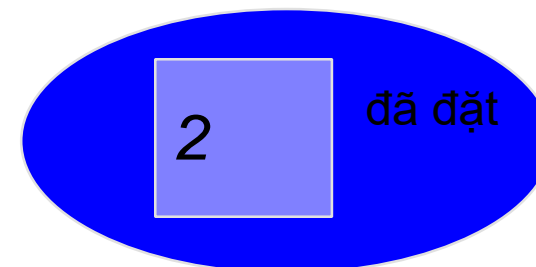
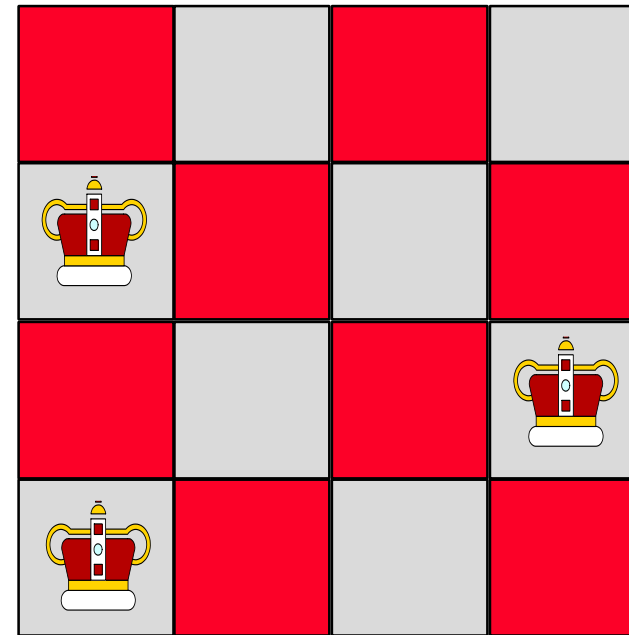
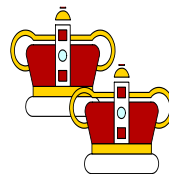
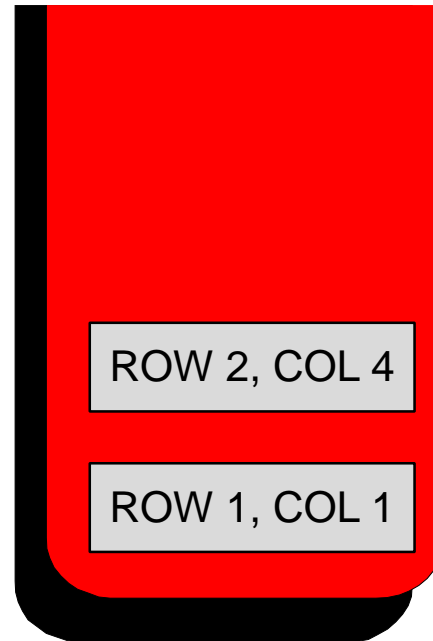
Thuật toán làm việc như thế nào

Xếp được con
hậu ở dòng 2 ta
tiếp tục xếp con
hậu ở dòng 3



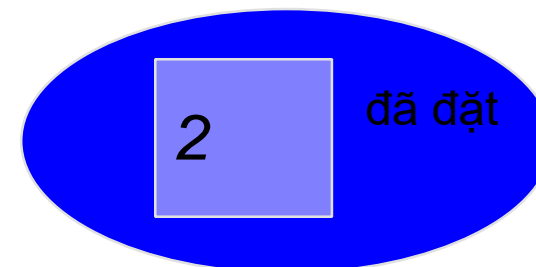
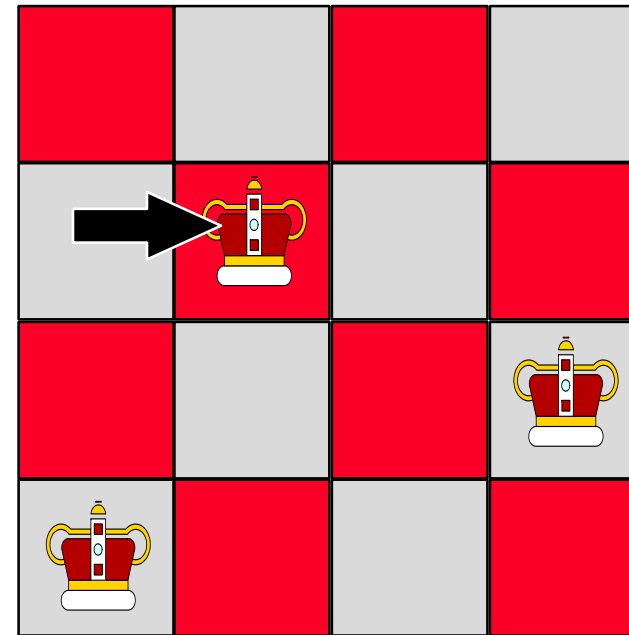
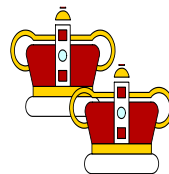
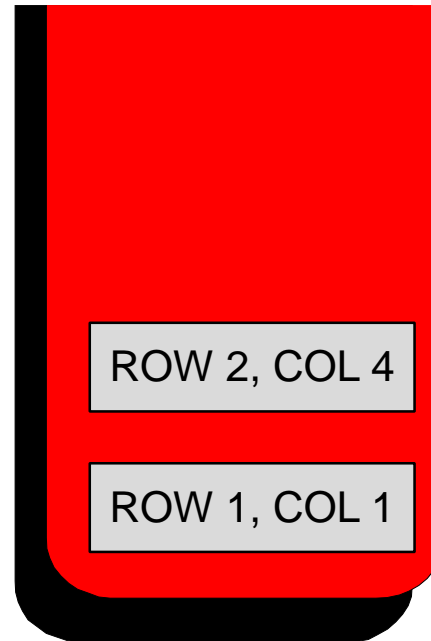
Thuật toán làm việc như thế nào

Thử xếp con hậu
ở dòng 3 vào cột
1



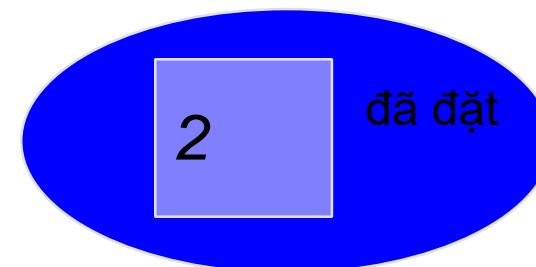
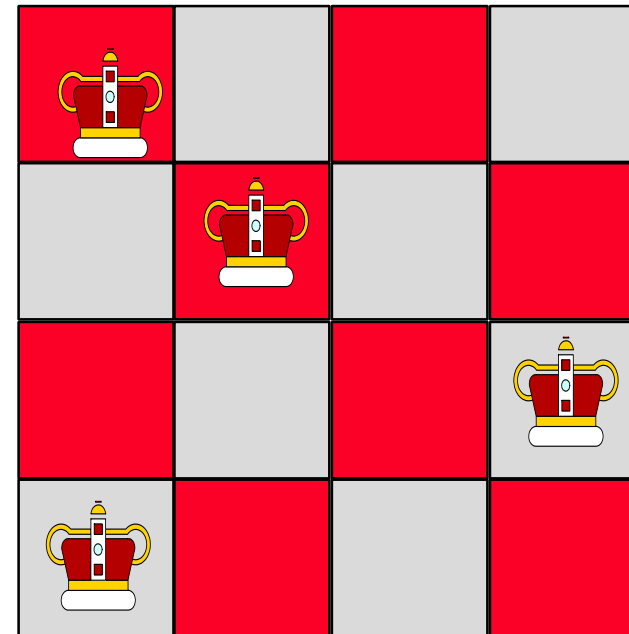
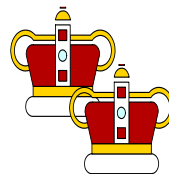
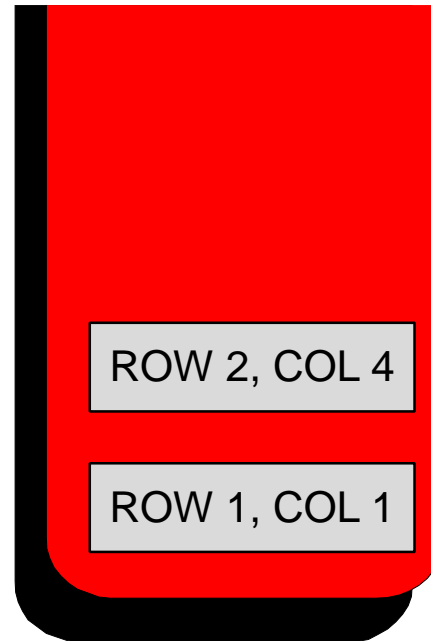
Thuật toán làm việc như thế nào

Thử xếp con hậu
ở dòng 3 vào cột
2



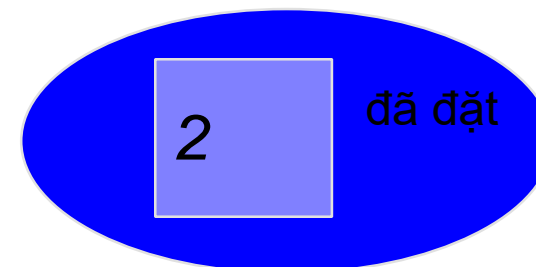
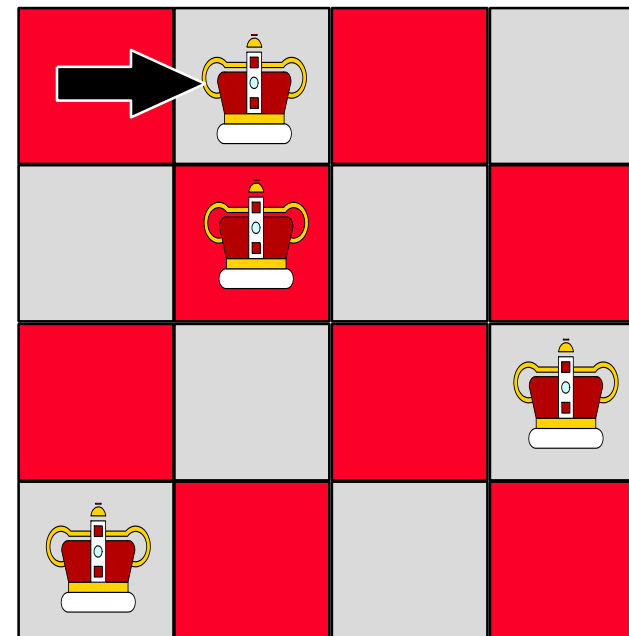
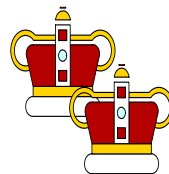
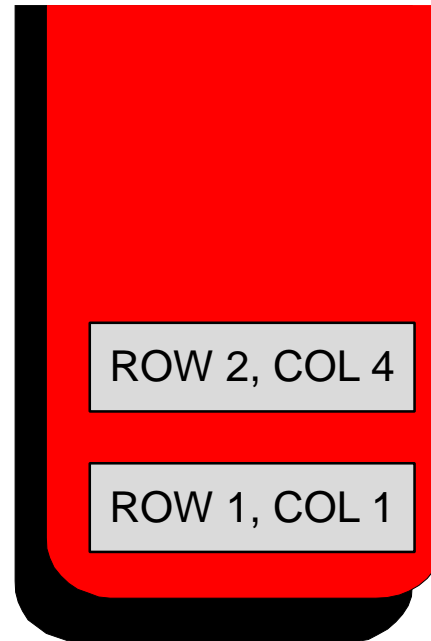
Thuật toán làm việc như thế nào

Xếp được con
hậu ở dòng 3 ta
tiếp tục xếp con
hậu ở dòng 4:
Thử cột 1



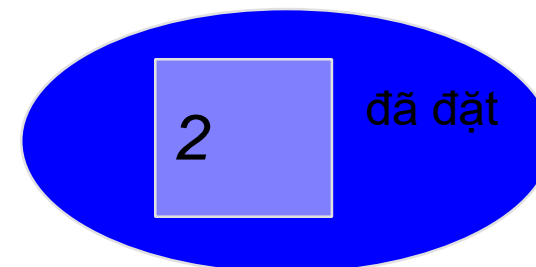
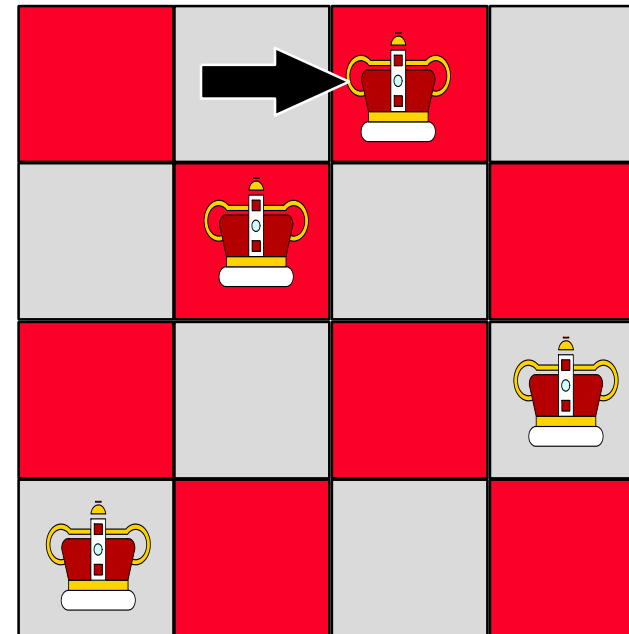
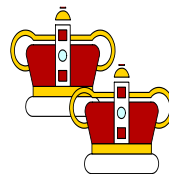
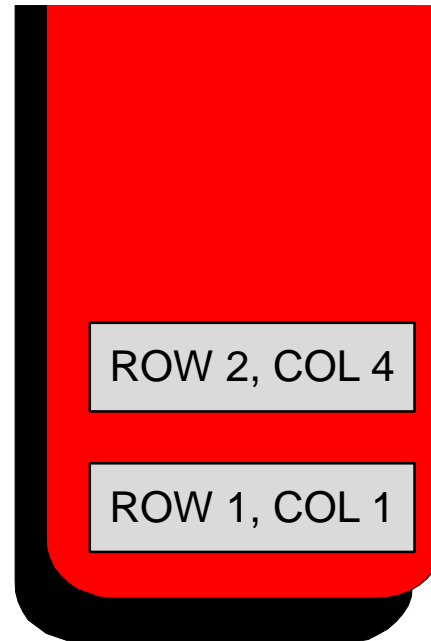
Thuật toán làm việc như thế nào

Thử xếp được
con hậu ở dòng
4 vào cột 2



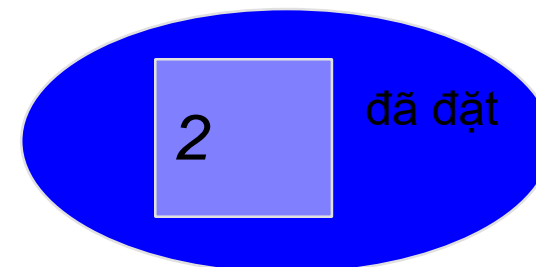
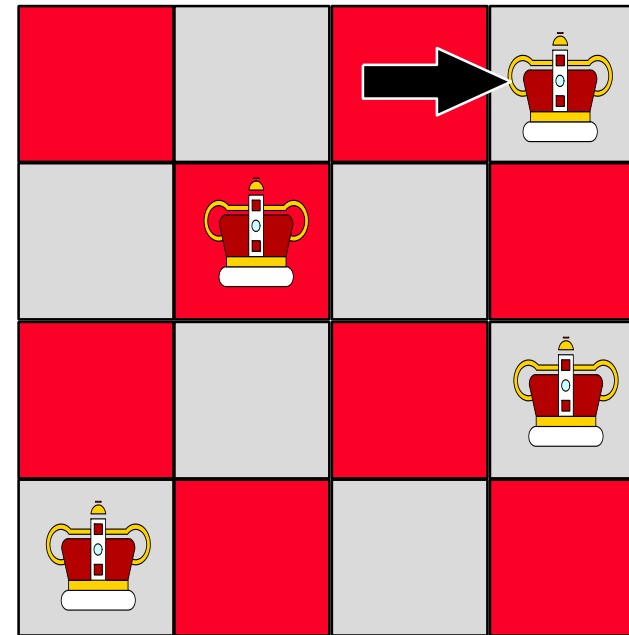
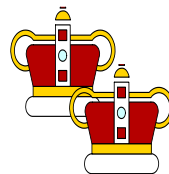
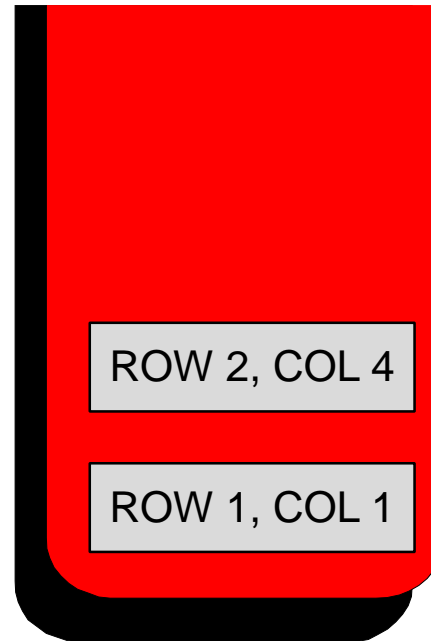
Thuật toán làm việc như thế nào

Thử xếp được
con hậu ở dòng
4 vào cột 3



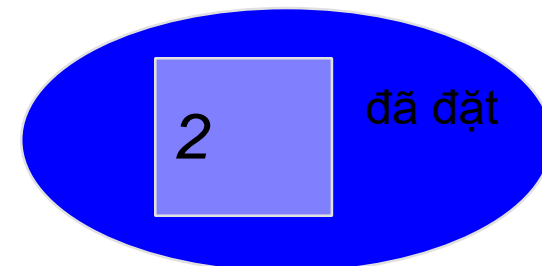
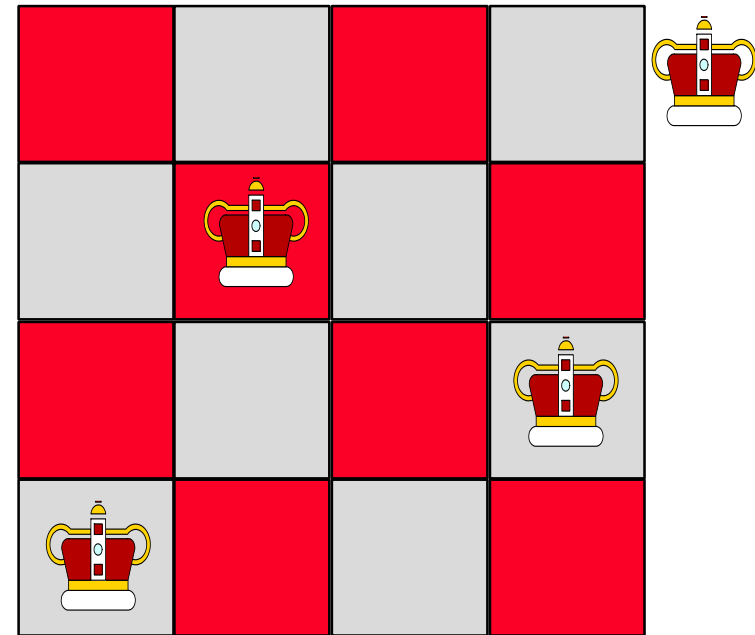
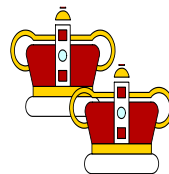
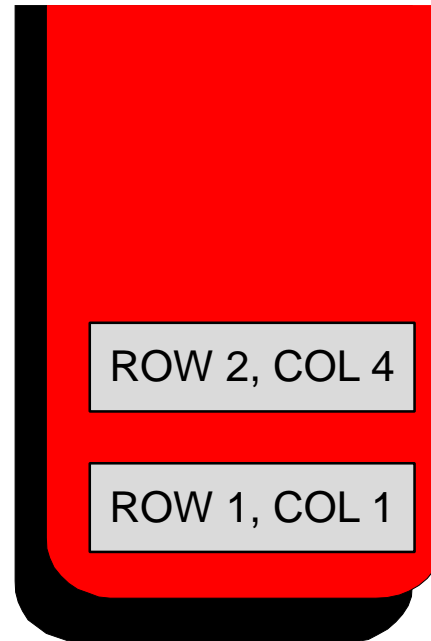
Thuật toán làm việc như thế nào

Thử xếp được
con hậu ở dòng
4 vào cột 4



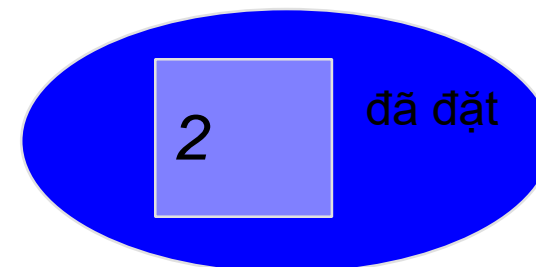
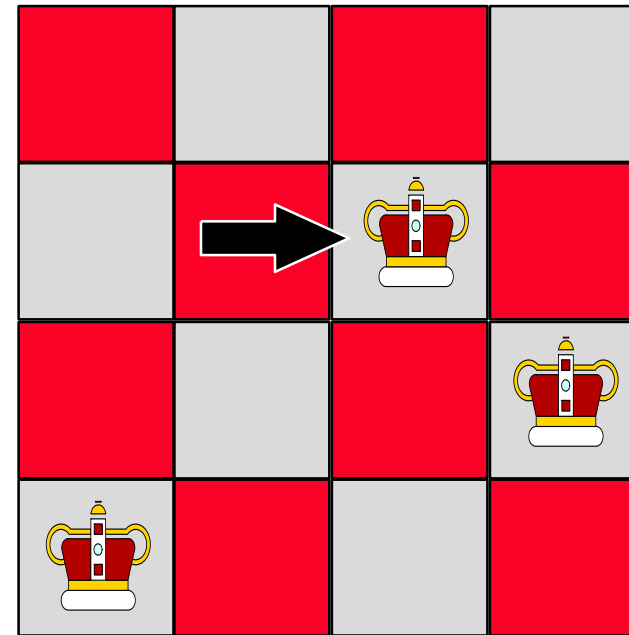
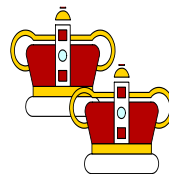
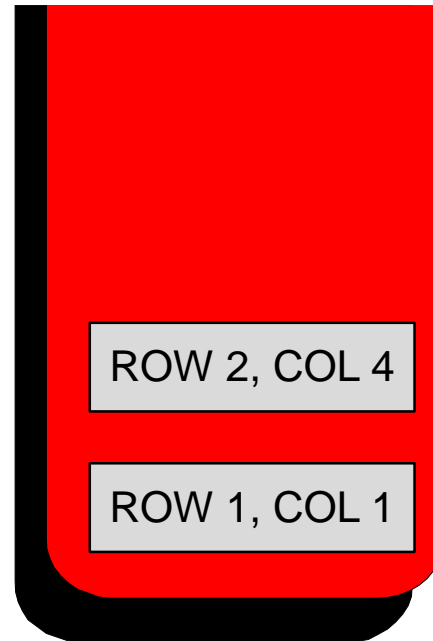
Thuật toán làm việc như thế nào

Không xếp được
con hậu ở dòng
4



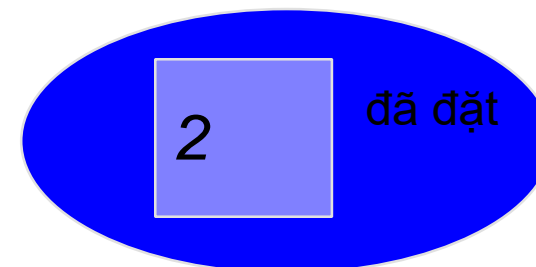
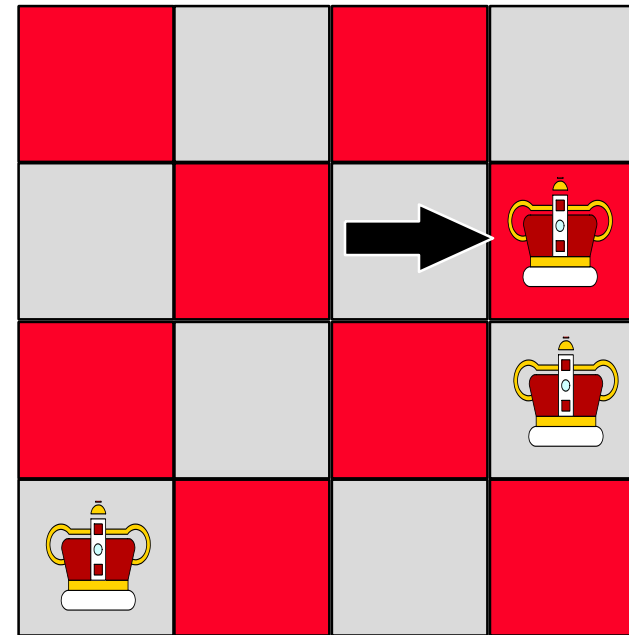
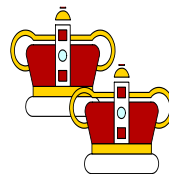
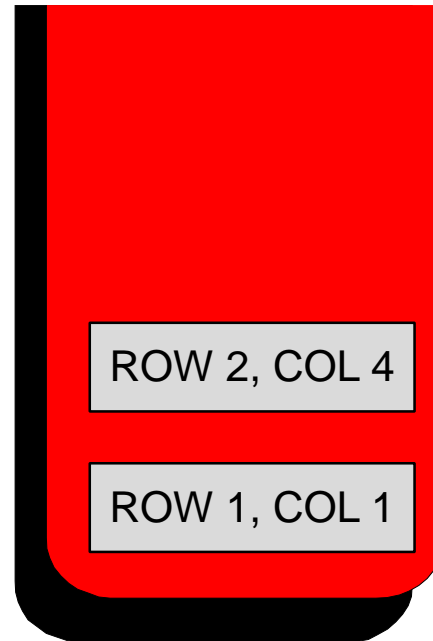
Thuật toán làm việc như thế nào

Quay lại tìm vị trí mới cho con hậu ở dòng 3: Thử cột 3



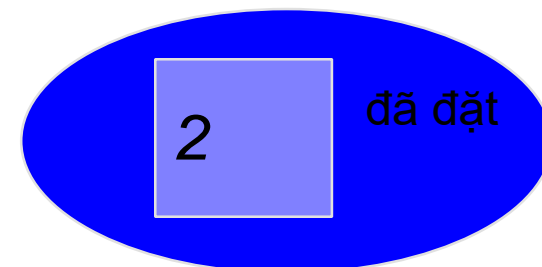
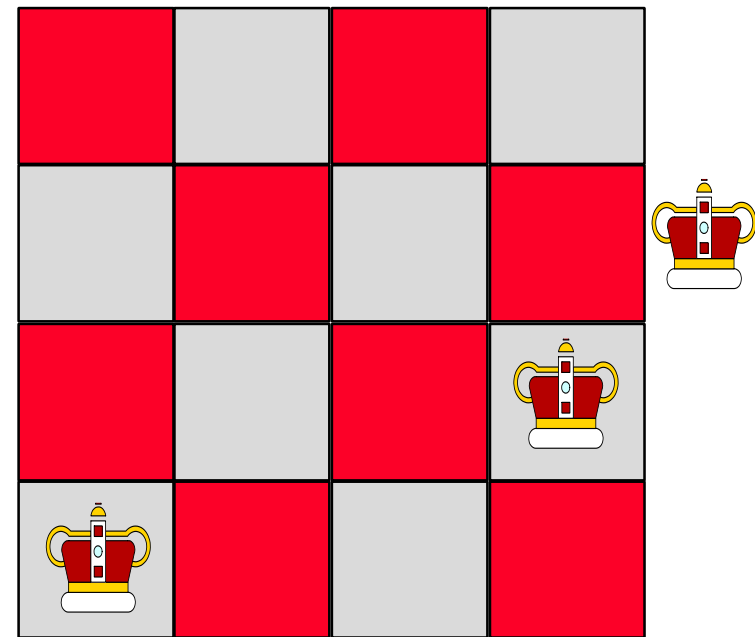
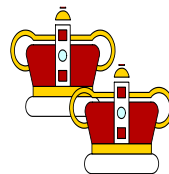
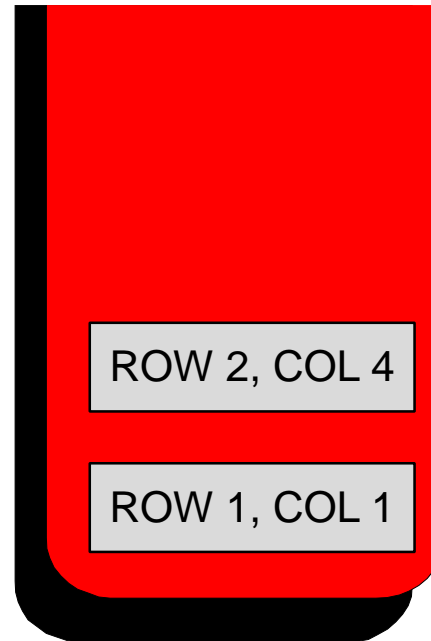
Thuật toán làm việc như thế nào

Quay lại tìm vị trí mới cho con hậu ở dòng 3: Thử cột 4



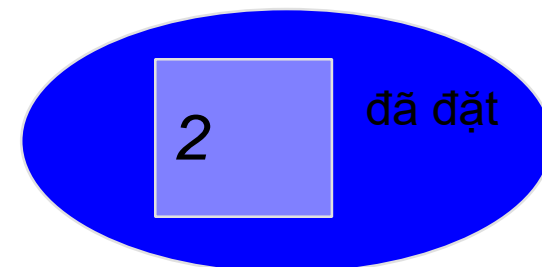
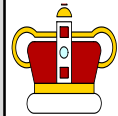
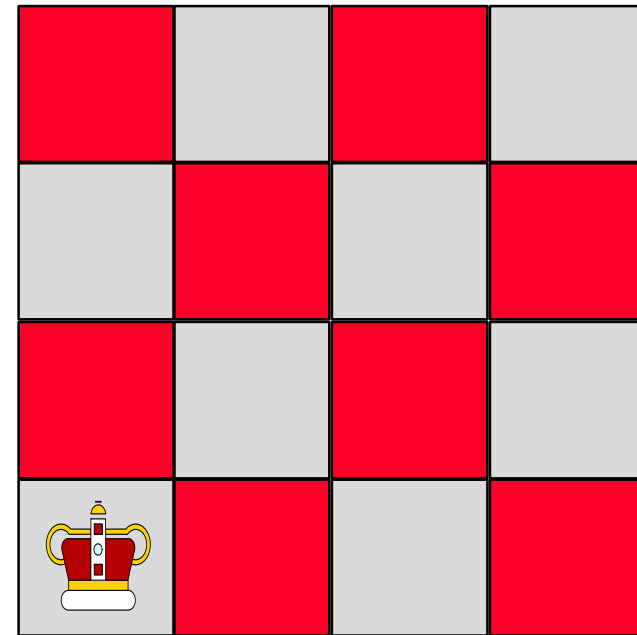
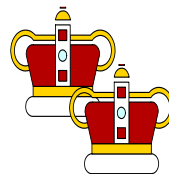
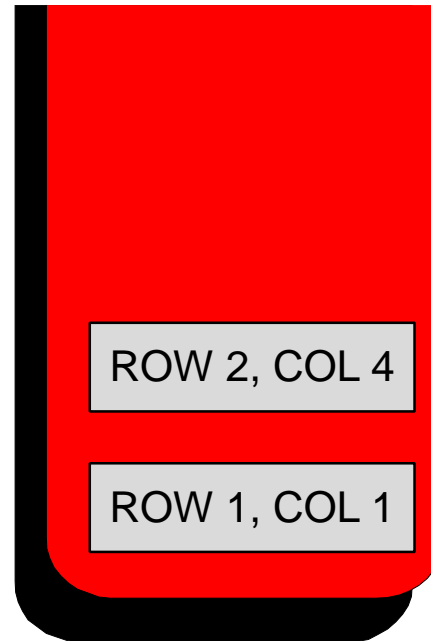
Thuật toán làm việc như thế nào

Không có cách
xếp mới cho con
hậu ở dòng 3



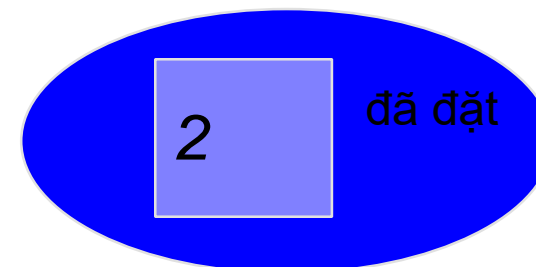
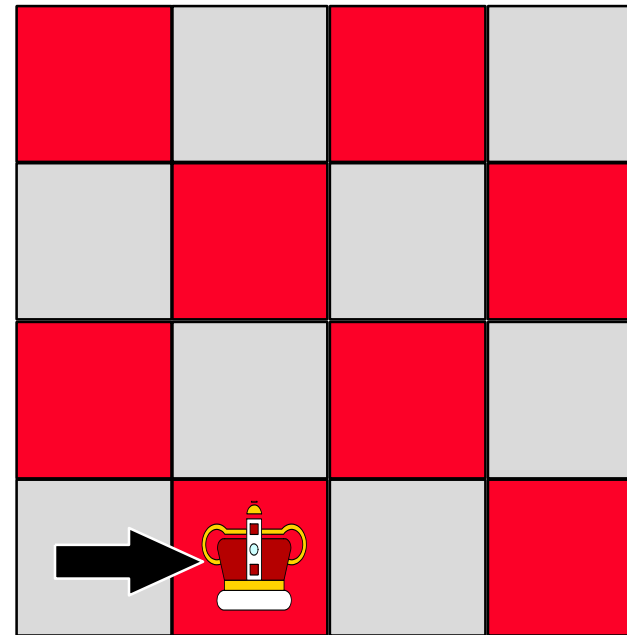
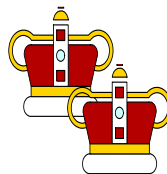
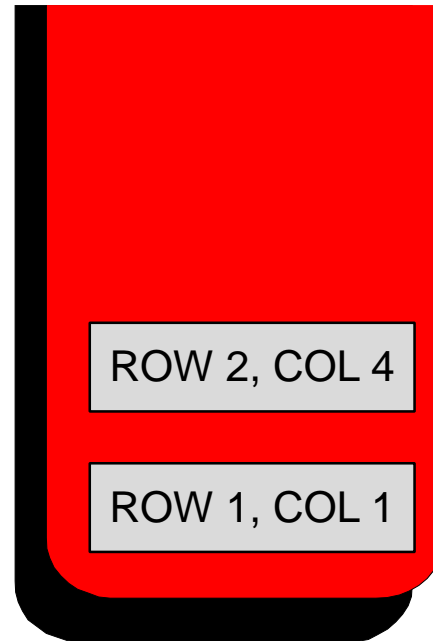
Thuật toán làm việc như thế nào

Quay lại tìm cách
xếp mới cho con
hậu ở dòng 2:
Không có

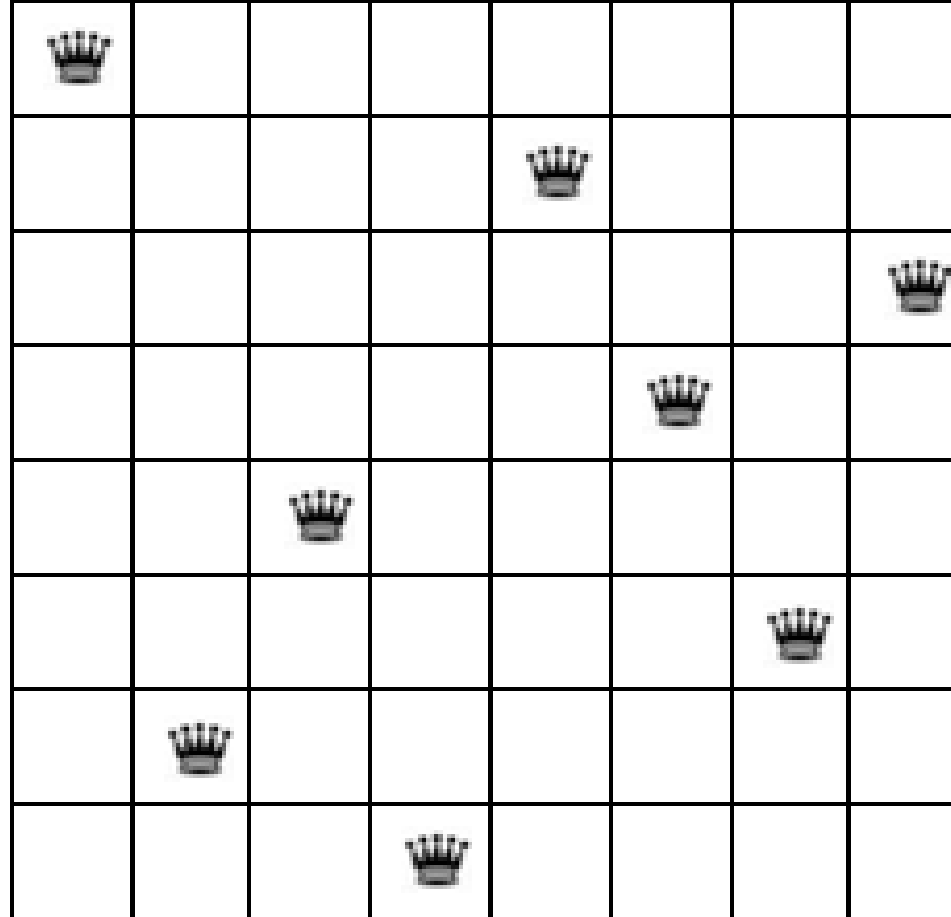


Thuật toán làm việc như thế nào

Quay lại tìm cách
xếp mới cho con
hậu ở dòng 1:
Chuyển sang cột
2



Một lời giải của bài toán xếp hậu khi $n = 8$



Phương pháp Liệt kê các m-tập con của n-tập

- **Bài toán:** Liệt kê các tập con m phần tử của tập $N = \{1, 2, \dots, n\}$.

- Bài toán dẫn về: Liệt kê các phần tử của tập:

$$S(m, n) = \{ (x_1, \dots, x_m) \in N^m : 1 \leq x_1 < \dots < x_m \leq n \}$$

Giải quyết 2 vấn đề mẫu chốt

- Từ điều kiện: $1 \leq a_1 < a_2 < \dots < a_m \leq n$

suy ra $S_1 = \{1, 2, \dots, n-(m-1)\}$.

- Giả sử đã có tập con (a_1, \dots, a_{k-1}) .

Từ điều kiện $a_{k-1} < a_k < \dots < a_m \leq n$, ta suy ra

$$S_k = \{a_{k-1}+1, a_{k-1}+2, \dots, n-(m-k)\}.$$

- Để cài đặt vòng lặp liệt kê các phần tử của S_k , dễ thấy là ta có thể sử dụng vòng lặp

for (y=a[k-1]+1;y<=n-m+k;y++) ...

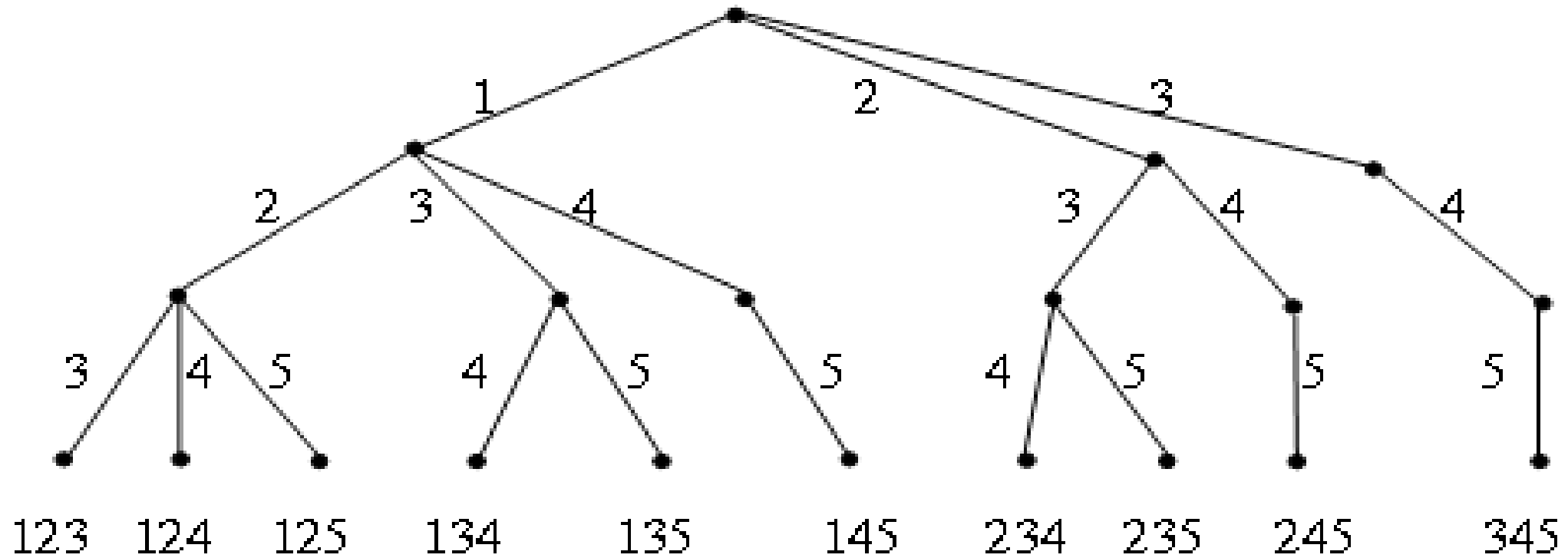
Code C (dung vòng lặp)

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int n, m, count;
int a[20];
void Ghinhan() {
    int i, j;
    count++;
    printf("Tap con thu %i. ",count);
    for (i=1 ; i<= m ;i++) {
        j=a[i];
        printf("%i ",j);
    }
    printf("\n");
}
```

```
void MSet(int i){
    int j;
    for (j = a[i-1] +1; j<= n-m+i; j++) {
        a[i] = j;
        if (i==m) Ghinhan();
        else MSet(i+1);
    }
}

int main() {
    printf("n, m = ");
    scanf("%i %i",&n, &m); printf("\n");
    a[0]=0; count = 0; MSet(1);
    printf("Count = %d ", count);
    getch();
}
```

Cây liệt kê $S(5,3)$



Thuật toán liệt kê các tập con k phần tử (đệ quy)

Thuật toán quay lui liệt kê các tập con k phần tử:

```
void Try( int i)
{ int j;
  For (j = x[i-1] + 1; j<= n-k+i; j++)
  {
    x[i] = j ;
    If (i == k) Printresult
    Else Try(i+1);
  }
}
```

Bài toán xếp hậu: Thuật toán quay lui (đệ quy)

Thủ tục đặt hậu:

```
void Try(int i)
```

```
{  int j;
```

```
    for (j = 1;j<= n;j++)
```

```
        if (a[j] and b[i + j] and c[i - j])
```

```
            { x[i] = j;
```

```
                if (i == n) PrintResult
```

```
                else
```

```
                {          a[j] = False; b[i + j] = False; c[i - j] = False;
```

```
                          Try(i + 1);
```

```
                          a[j] = True; b[i + j] = True; c[i - j] = True;
```

```
                }
```

```
            }
```

```
}
```