Bài 2.3: Bài toán xếp quân cờ

1. ***Xếp quân hậu***
2. Bài toán:

Xét bàn cờ tổng quát kích thước nxn. Một quan hậu trên bàn cờ có thể ăn được các quân khác tại các ô cùng hàng, cùng cột hoặc cùng đường chéo. Hãy tìm cách xếp n quân hậu trên bàn cờ sao cho không quân nào ăn quân nào.

1. Phân tích:

* Do quân hậu ăn được các ô cùng hàng nên n quân hậu sẽ được đặt mỗi con 1 hàng. Gọi quân hậu ở hàng 1 là quân hậu 1, … , quân hậu ở hàng n là quân hậu n. Nghiệm của bài toán là vị trí cột của những quân hậu.
* Theo định hướng Đông(Phải), Tây(Trái), Nam(Dưới), Bắc(Trên) ta thấy:
* Một đường chéo hướng ĐB – TN (gọi là đường chéo ngược) có tính chất Hàng + Cột = C = const
* Một đường chéo hướng ĐN – TB (gọi là đường chéo xuôi ) có tính chất

Hàng – Cột = C

* Số đường chéo xuôi = Số đường chéo ngược = SoDuongCheo = 2\*n - 1
* Ta có 3 mảng đánh dấu:
* Mảng cot[1..n]: cot[j] = 1 nếu cột j còn tự do, cot[j] = 0 nếu cột j đã bị 1 quân hậu ăn
* Mảng cheonguoc[1…SoDuongCheo](tính từ góc trái từ trên xuống): cheonguoc[i + j - 1] = 1 nếu đường chéo thuộc cột thuộc cột j và hàng i còn tự do, ngược lại = 0 nếu nó bị 1 quân hậu khống chế
* Mảng cheoxuoi[1..SoDuongCheo] (tính từ góc trái từ dưới lên):

cheoxuoi[ i – j + SoDuongCheo] = 1 nếu đường chéo thuộc cột j và hàng i còn tự do, ngược lại = 0 nếu nó bị 1 quân hậu khống chế.

1. Thuật toán

Sử dụng thuật toán quay lui

* Xét các cột, thử đặt quân hậu 1 vào 1 cột, với mỗi cách đặt như vậy, xét tất cả các cách đặt quân hậu 2 ko bị quân hậu 1 ăn, thử lại 1 cách đặt và xét tiếp các cách đặt quân hậu 3…
* Trước khi chọn cột j cho quân hậu thứ i thì ta phải chọn ô(i, j) sao cho cột[j] = 0, cheonguoc[i+j-1] = 0, cheoxuoi[i – j + SoDuongCheo] = 0 để đảm bảo quân hậu ko bị những quân hậu đặt trước đó ăn.
* Lược đồ: Chú ý: 1: còn trống, 0: ko còn trống

Try(i)

for(j = 1 .. n) //Duyệt qua số cột có thể đặt hậu if(cot[j] = 1 & cheonguoc[i+j-1] = 1 & cheoxuoi[i – j + SoDuongCheo] = 1) // cột j chấp nhận được pa[i] = j; // đặt quân hậu i tại cột j // đánh dấu cột, đường chéo xuôi, chéo ngược mà nó đi qua không còn trống nữa

cottrong[i] = 0; cheoxuoi[j-i+KICHTHUOC-1] = 0; cheonguoctrong[j+i] = 0;

if(i = n) // nếu đặt quân hậu cuối cùng inda();

else

Try(i+1);

//khôi phục trạng thái trước khi đặt quân hậu i

cottrong[i] = 1;

cheoxuoitrong[j-i+KICHTHUOC-1] = 1;

cheonguoctrong[j+i] = 1;

endif

endfor

End.

1. ***Xếp quân mã***
2. Bài toán:

Hãy chỉ ra một cách đi của con mã để xuất phát từ 1 ô cờ (x, y) nào đó (x, y= 1..8) có thể đi qua tất cả các ô cờ, mỗi ô duy nhất 1 lần

1. Phân tích:

* Do mỗi quân mã có thể đi hình chữ L, nên tại mỗi vị trí quân mã có 8 khả năng đi.Tương ứng với 8 cách đi thì số hàng và cột của ô (x, y) thay đổi như sau:
* Tiến 2 rồi phải 1: ô mới là (x + 2, y + 1)
* Tiến 2 rồi trái 1: ô mới là (x+2, y - 1)
* Lùi 2 rồi phải 1: ô mới là (x – 2, y + 1)
* Lùi 2 rồi trái 1: ô mới là (x – 2, y - 1)
* Phải 2 rồi tiến 1: ô mới là (x + 1, y + 2)
* Phải 2 rồi lùi 1: ô mới là (x – 1, y + 2)
* Trái 2 rồi tiến 1: ô mới là (x + 1, y - 2)
* Trái 2 rồi lùi 1: ô mới là (x – 1, y - 2)
* Mỗi lần đi của quân mã phải thỏa mãn: các ô đến còn nằm trong bàn cờ và còn trống(để thỏa mãn mỗi ô chỉ đi 1 lần)
* Biểu diễn bàn cờ = Ma trận kích thước 8\*8
* Ban đầu mọi ô đều trống

1. Thuật toán

* Sử dụng thuật toán quay lui
* Phương án gồm (x1, …, x64) là 64 ô trên bàn cờ 8\*8
* Mỗi nhận 1 ô cụ thể. Xây dựng thành phần :

+ Thỏa mãn là 1 trong 8 khả năng đi của quân mã

+ Ô định đi vẫn trống

+ Ô nằm trên bàn cờ

* Đánh dấu ô này đã đi qua
* Nếu đi được đủ hết các ô thì in ra kết quả ngược lại chưa đủ thì đi tiếp các bước tiếp theo

Lược đồ

Try(int k, int x, int y) // nước đi thứ k, xuất phát từ ô (x, y)

// v thuộc khả năng thành phần nghiệm x

for(i=1..8) //8 bước đi có thể của quân mã

<xmoi, ymoi>;

if(xmoi >=1 &&xmoi <= n && ymoi >=1 && ymoi <= n && M[xmoi][ymoi] == 0) // ô thoa mãn và còn trống

M[xmoi][ymoi] = k; // đánh dấu ô này đã được đi qua

if(k = n\*n) // neu di het cac o

<chapnhankq>;

else

Try(k+1, xmoi, ymoi);//Tìm nước đi tiếp theo

M[xmoi][ymoi] = 0; // đánh dấu lại, trc khi di nuoc thu k

endif

endfor

End.