INSTITUTO POLITÉCNICO NACIONAL (UPIIZ)

Caballo

(14 noviembre 2021)

Javier Alejandro Macias Basurto 3CM3

I. Introducción

El problema del caballo es un antiguo problema matemático en el que se pide que, teniendo una cuadrícula de n x n casillas y un caballo de ajedrez colocado en una posición cualquiera (x, y), el caballo pase por todas las casillas y una sola vez.

Muchos matemáticos han buscado una solución matemática a este problema, entre ellos Leonhard Euler.

Se han encontrado muchas soluciones a este problema y de hecho no se sabe con seguridad de cuántas maneras diferentes es posible solucionarlo.

Algunas variaciones de este problema han sido estudiadas por los matemáticos, tales como:

- Buscar soluciones cíclicas, en la cual se debe llegar a la misma casilla de la cual se partió.
- Tableros de diferente número de columnas o diferente número de filas.
- Juegos de dos jugadores basados en la idea.
- Problemas usando ligeras variaciones en la forma de moverse el caballo.

El problema del caballo es una forma del problema más general problema de la ruta Hamiltoniana en la teoría de grafos.

II. Desarrollo

Este algoritmo trata en resumen de colocar inicialmente el caballo en una casilla de un tablero de ajedrez y moverlo hasta haber pisado todas las casillas sin haber estado más de una vez en la misma casilla. En la figura 1 se muestra un ejemplo del movimiento del caballo. En la figura 2 se muestra un ejemplo de los ocho movimientos iniciales posibles de un caballo en un tablero de 8x8.

Figura 1. Movimiento del Caballo.

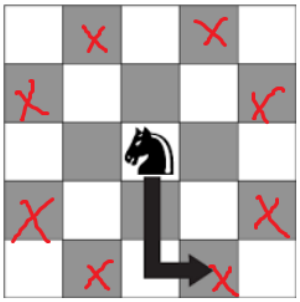

Figura 2. Los 8 movimientos de un caballo.

Para hacer esto utilizaremos un algoritmo recursivo de vuelta atrás (backtracking) basado en ir colocando, según un orden, un caballo detrás de otro evitando las casillas ya ocupadas.

El tablero lo guardaremos como una matriz de n x n (siendo n el número de filas y columnas) de tal forma que inicialmente tendrá valores cero (casillas todavía no visitadas). El programa recibe como único parámetro la casilla del tablero de donde el caballo empieza caballo, donde se ubicará el primer caballo marcándolo con un 1 en el tablero para indicar esta circunstancia.

A partir de la posición de arranque del caballo, habría ocho movimientos posibles para poner un segundo caballo como se puede ver en la figura 2 ya que los movimientos son en L. El orden en el que se exploraran los primeros moviemientos será:

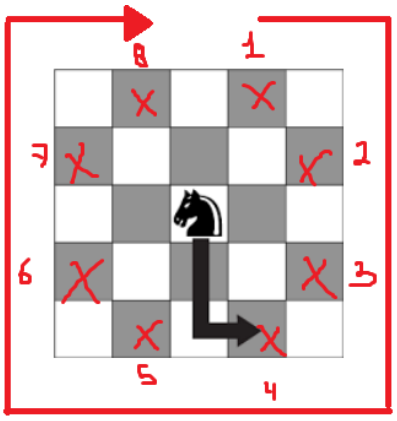

Figura 3. Orden primeros Movimientos.

Marcaremos con un 1 la casilla ya usada y con un cero la libre. Y lo guardaremos en un historial de casillas para después verificar si el caballo ya estuvo en esa casilla.

Para iniciar el recorrido necesitamos la posición inicial de recorrido del caballo en el que empezaremos en 0x0 y un número n que será nuestro tablero lo guardaremos como una matriz de n x n (siendo n el número de filas y columnas) después empezaremos con los movimientos, evaluando movimientos hacia arriba, abajo, izquierda y derecha

```
public void iniciarRecorrido(int x, int y, int n){
    this.n=n;
    caballo= new Caballo(x,y);
    tablero= new int[n][n];
    inicializarTablero();
    //int[]aux3;
    //posicionar caballo dentro del tablero
    tablero[x][y]=1; /* 1= casilla usada, 0=casilla libre */
    //iniciar recorrido de casillas
    historialCasillas.add(new int[]{x,y});

    for(int i=0;i<(n*n)-1;i++){
       //Evaluar sigiente casilla
       menor=new int[2];
       movM=9;

       evaluarArriba();
       evaluarDerecha();
       evaluarAbajo();
       evaluarIzquierda();

        historialCasillas.add(menor);
        tablero[menor[0]][menor[1]]=1;
        caballo.setX(menor[0]);
        caballo.setY(menor[1]);
        System.out.println();

    }
    System.out.println(historialCasillas.size());
    for(int j=0;j<historialCasillas.size();j++){
        System.out.println("Posicion "+j+1+": "+historialCasillas.get(j)[0]+","+historialCasillas.get(j)[1]);
    }
}
```

Figura 4. Iniciar recorrido.

```
public void evaluarArriba(){
    /* **********  Evaluar arriba  ********** */
        if (caballo.getY()-2>=0){ //si no esta en las primeras dos filas evaluar hacia arriba
            System.out.println();
            if(caballo.getX()-1>=0){//evaluar a la izquierda
                System.out.println();
                if(tablero[caballo.getX()-1][caballo.getY()-2]==0){
                    System.out.println();
                    aux[0]=caballo.getX()-1;
                    aux[1]=caballo.getY()-2;
                    aux1= movimientosDisponibles(aux);
                    if(aux1<movM){
                        System.out.println();
                        movM=aux1;
                        menor[0]=caballo.getX()-1;
                        menor[1]=caballo.getY()-2;
                    }
                    System.out.println();
                }
            }
            if(caballo.getX()+1<=n-1){//evaluar a la derecha
                System.out.println();
                if(tablero[caballo.getX()+1][caballo.getY()-2]==0){
                    System.out.println();
                    aux[0]=caballo.getX()+1;
                    aux[1]=caballo.getY()-2;
                    aux1= movimientosDisponibles(aux);
                    if(aux1<movM){
                        System.out.println();
                        movM=aux1;
                        menor[0]=caballo.getX()+1;
                        menor[1]=caballo.getY()-2;
                    }
                    System.out.println();
                }
```

Figura 5. Evaluar Arriba.

```
public void evaluarDerecha(){
    /* **********  Evaluar derecha  ********** */
        if (caballo.getX()+2<=n-1){ //si no esta en las dos ultimas columnas a la derecha evaluar hacia la derecha
            System.out.println();
            if(caballo.getY()-1>=0){//evaluar arriba
                System.out.println();
                if(tablero[caballo.getX()+2][caballo.getY()-1]==0){
                    System.out.println();
                    aux[0]=caballo.getX()+2;
                    aux[1]=caballo.getY()-1;
                    aux1= movimientosDisponibles(aux);
                    if(aux1<movM){
                        System.out.println();
                        movM=aux1;
                        menor[0]=caballo.getX()+2;
                        menor[1]=caballo.getY()-1;
                        System.out.println();
                    }
                }
            }
            if(caballo.getY()+1<=n-1){//evaluar abajo
                System.out.println();
                if(tablero[caballo.getX()+2][caballo.getY()+1]==0){
                    System.out.println();
                    aux[0]=caballo.getX()+2;
                    aux[1]=caballo.getY()+1;
                    aux1= movimientosDisponibles(aux);
                    if(aux1<movM){
                        System.out.println();
                        movM=aux1;
                        menor[0]=caballo.getX()+2;
                        menor[1]=caballo.getY()+1;
                    }
                    System.out.println();
                }
            }
        }
```

Figura 6. Evaluar Derecha.

```java
public void evaluarAbajo(){
    /* **********  Evaluar abajo  ********** */
        if (caballo.getY()+2<=n-1){ //si no esta en las ultimas dos filas evaluar hacia abajo
            System.out.println();
            if(caballo.getX()+1<=n-1){//evaluar a la derecha
                System.out.println();
                if(tablero[caballo.getX()+1][caballo.getY()+2]==0){
                    System.out.println();
                    aux[0]=caballo.getX()+1;
                    aux[1]=caballo.getY()+2;
                    aux1= movimientosDisponibles(aux);
                    if(aux1<movM){
                        System.out.println();
                        movM=aux1;
                        menor[0]=caballo.getX()+1;
                        menor[1]=caballo.getY()+2;
                    }
                    System.out.println();
                }
            }
            if(caballo.getX()-1>=0){//evaluar a la izquierda
                System.out.println();
                if(tablero[caballo.getX()-1][caballo.getY()+2]==0){
                    System.out.println();
                    aux[0]=caballo.getX()-1;
                    aux[1]=caballo.getY()+2;
                    aux1= movimientosDisponibles(aux);
                    if(aux1<movM){
                        System.out.println();
                        movM=aux1;
                        menor[0]=caballo.getX()-1;
                        menor[1]=caballo.getY()+2;
                    }
                    System.out.println();
                }
            }
```

Figura 7. Evaluar Abajo.

```java
public void evaluarIzquierda(){
    /* **********  Evaluar izquierda  ********** */
        if (caballo.getX()-2>=0){ //si no esta en las primeras dos fcolumnas evaluar hacia la izquierda
            System.out.println();
            if(caballo.getY()+1<=n-1){//evaluar hacia abajo
                System.out.println();
                if(tablero[caballo.getX()-2][caballo.getY()+1]==0){
                    System.out.println();
                    aux[0]=caballo.getX()-2;
                    aux[1]=caballo.getY()+1;
                    aux1= movimientosDisponibles(aux);
                    if(aux1<movM){
                        System.out.println();
                        movM=aux1;
                        menor[0]=caballo.getX()-2;
                        menor[1]=caballo.getY()+1;
                    }
                    System.out.println();
                }
            }
            if(caballo.getY()-1>=0){//evaluar hacia arriba
                System.out.println();
                if(tablero[caballo.getX()-2][caballo.getY()-1]==0){
                    System.out.println();
                    aux[0]=caballo.getX()-2;
                    aux[1]=caballo.getY()-1 ;
                    aux1= movimientosDisponibles(aux);
                    if(aux1<movM){
                        System.out.println();
                        movM=aux1;
                        menor[0]=caballo.getX()-2;
                        menor[1]=caballo.getY()-1;
                    }
                    System.out.println();
                }
```

Figura 8. Evaluar Izquierda.

Después se imprime en consola el orden de las casillas visitadas.

```
64
Matriz:8x8
Posicion 1: 0,0
Posicion 2: 2,1
Posicion 3: 4,0
Posicion 4: 6,1
Posicion 5: 7,3
Posicion 6: 6,5
Posicion 7: 7,7
Posicion 8: 5,6
Posicion 9: 7,5
Posicion 10: 6,7
Posicion 11: 4,6
Posicion 12: 2,7
Posicion 13: 0,6
Posicion 14: 1,4
Posicion 15: 0,2
Posicion 16: 1,0
Posicion 17: 3,1
Posicion 18: 5,0
Posicion 19: 7,1
Posicion 20: 6,3
Posicion 21: 4,2
Posicion 22: 3,0
Posicion 23: 1,1
Posicion 24: 0,3
Posicion 25: 2,2
Posicion 26: 0,1
Posicion 27: 2,0
Posicion 28: 1,2
Posicion 29: 0,4
Posicion 30: 2,3
Posicion 31: 1,5
Posicion 32: 0,7
Posicion 33: 2,6
Posicion 34: 3,4
Posicion 35: 1,3
Posicion 36: 0,5
Posicion 37: 1,7
Posicion 38: 2,5
Posicion 39: 3,7
Posicion 40: 1,6
Posicion 41: 3,5
Posicion 42: 4,7
Posicion 43: 6,6
Posicion 44: 5,4
Posicion 45: 3,3
Posicion 46: 4,1
Posicion 47: 6,0
Posicion 48: 5,2
Posicion 49: 4,4
Posicion 50: 3,2
Posicion 51: 5,1
Posicion 52: 7,0
Posicion 53: 6,2
Posicion 54: 7,4
Posicion 55: 5,3
Posicion 56: 7,2
Posicion 57: 6,4
Posicion 58: 7,6
Posicion 59: 5,7
Posicion 60: 4,5
Posicion 61: 2,4
Posicion 62: 4,3
Posicion 63: 5,5
Posicion 64: 3,6
```

Figura 9. Resultados.

III. Conclusión

Este Algoritmo es un problema de matemáticas discretas el cual pone a prueba nuestras capacidades para resolverlo, podemos pensar muchas soluciones para resolverlo algunas mejores o más optimizadas pero con que todas lleguen a su cometido, este problema es muy antiguo y surgió de encontrar esta solución a lo que llamaríamos tanteo lo que es prácticamente imposible por lo que hasta el año de 1759 Euler un profesor de matemáticas encontró una solución, pero en el software podemos hacer un algoritmo a base de recursividad ,condiciones y evaluaciones.

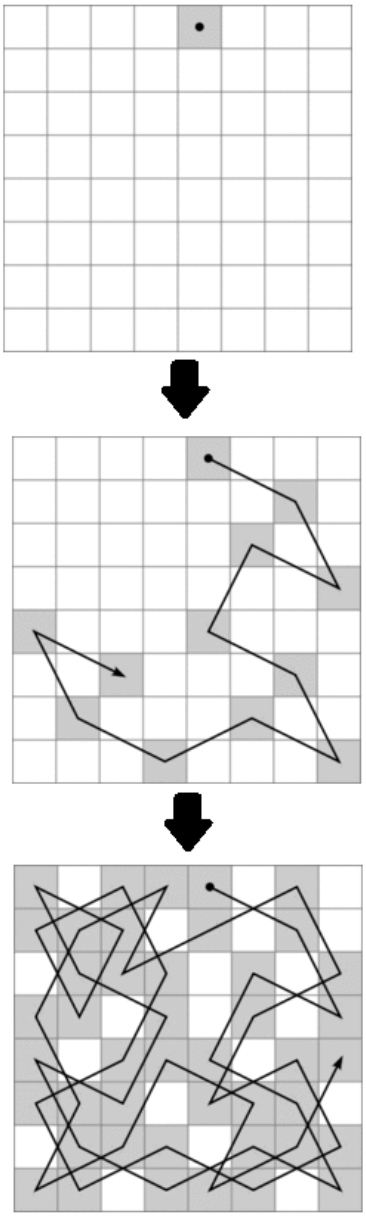

Figura 10. Solucion Caballo.

IV. Referencias

colaboradores de Wikipedia. (2021e, abril 24). Problema del caballo. Wikipedia, la enciclopedia libre. Recuperado 14 de noviembre de 2021, de https://es.wikipedia.org/wiki/Problema_del_caballo#:%7E:text=El%20problema%20del%20caballo%20es,casillas%20y%20una%20sola%20vez.

Palazzesi, A. (2018, 7 noviembre). El problema del caballo. NeoTeo. Recuperado 14 de noviembre de 2021, de https://www.neoteo.com/el-problema-del-caballo/