UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA CAMPUS ZACATECAS

ANALISIS DE ALGORITMOS

CRUZ LEIJA ROBERTO OSWALDO

ROBERTO COVARRUBIAS LUNA

PROBLEMA DEL CABALLO

Introducción:

El llamado “Problema del caballo” es un antiguo problema matemático relacionado con el ajedrez. Consiste en encontrar una secuencia de movimientos -válidos- de esta pieza para que recorra todas las casillas del tablero, visitando cada una solo una vez. Verdaderos ejércitos de matemáticos han encarado este problema, pero sigue sin conocerse el número exacto de soluciones que existe. El problema ha sido planteado para tableros de diferentes tamaños y distintas condiciones iniciales, y sigue siendo tan atractivo como hace 1200 años

Reglas:

1. Movimientos legales (Solo en L).
2. No salirse del tablero.
3. Pasar solamente 1 ves por cada casilla.
4. Pasar por todas las casillas.

Ejemplo:

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

Código:

```
/*
 * To change this license header, choose License Headers in Project Properti
es.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author rober_xz5594r
 */
public class caballo
{
    public int [][] tablero;
    public int mx[]={-2,-1,1,2,2,1,-1,-2};
    public int my[]={1,2,2,1,-1,-2,-2,-1};
    boolean exito=true;


    public caballo()
    {
        this.tablero=new int[8][8];
    }


    public void resolver()
    {
        int px=5, py=0;
        int pxaux=0, pyaux=0;
        int cont=1;
        int posi=9;
        int aux=0;

        this.tablero[px][py]=1;
        do
        {
            pxaux=px;
            pyaux=py;
            for(int i=0; i<8; i++)
            {
                if(validar(px, py, mx[i], my[i])==1)
                {

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

```
                    {
                        aux+=validar(px+mx[i], py+my[i], mx[j], my[j]);
                    }
                    if ((aux<posi && aux!=0) || (aux<posi && aux==0 && cont==63))
                    {
                        pxaux=px+mx[i];
                        pyaux=py+my[i];
                        posi=aux;
                    }
                    aux=0;
                }
            }
            if(posi==9)
            {
                exito=false;
            }
            else
            {
                cont++;
                this.tablero[pxaux][pyaux]=cont;
                posi=9;
                px=pxaux;
                py=pyaux;
            }
            System.out.println("px="+px+", py="+py);
            imprimirTablero();
        }while((exito==true) && cont!=64);


    }
    public int validar(int px, int py, int x, int y)
    {
        if(((px+x)>=0 && (px+x)<=7) && ((py+y)>=0 && (py+y)<=7))
        {
            if(tablero[px+x][py+y]==0)
            {
                return 1;
            }
            else
            {
                return 0;
            }
        }
        else
```

```
        {
            return 0;
        }
    }
    public void imprimirTablero ()
    {
        for(int i=0; i<8; i++)
        {
            for(int j=0; j<8; j++)
            {
                System.out.print(""+tablero[i][j]+"\t");
            }
            System.out.println();
        }
    }
    public static void main(String args[]){

        caballo T = new caballo();
         T.resolver();
   }

}
```

Código comentado:

https://github.com/Robertocovarrubias/CaballitoErria/blob/master/Caballote

Capturas de pantalla:

Inicio:

Se inicia en X=5, Y=5

```
px=6, py=7
0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0
0       0       0       0       0       1       0       0
0       0       0       0       0       0       0       2
0       0       0       0       0       0       0       0
```

Final:

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

Inicia en X=2, Y=3

```
px=0, py=4
0       0       0       0       2       0       0       0
0       0       0       0       0       0       0       0
0       0       0       1       0       0       0       0
0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0
```

Final:

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