

# Taller 11

Carlos Barón - Andrés Cocunubo

May 6, 2018

## 1. Descripción del Problema

Se desea que la ficha de ajedrez del caballo recorra todo el tablero pasando una única vez por cada posición.

## 2. Formalización

### (a) Entradas :

- i. Tablero de ajedrez: Sea una matriz  $T^{n \times n} | n \in \mathbb{N}$ , donde el valor de cada celda es 0 si no está visitado o 1 de lo contrario.
- ii. Posición inicial del caballo: Posición  $i \wedge j | 1 \leq i \leq n \wedge 1 \leq j \leq n$

### (b) Salidas

- i. Mensaje si es o no posible que la ficha recorra todo el tablero.
- ii. Secuencia de saltos: Secuencia  $S$  de tamaño  $n \times n$  y con una relación de equivalencia, donde  $S[i] \neq S[1, \dots, i-1] \wedge S[i+1, \dots, |S|]$ .

## 3. Algoritmos

---

**Algorithm 1** Calcular posiciones jugables

---

```
1: procedure CALCULARPOSICIONES( $x, y, n$ )
2:   let  $P$  be an empty list
3:   let  $V \leftarrow \{-2, -1, 1, 2\}$ 
4:   for  $i \leftarrow 1$  to 4 do
5:      $k \leftarrow x + V[i]$ 
6:     for  $j \leftarrow 1$  to 4 do
7:       if  $|V[i]| \neq |V[j]|$  then
8:          $l \leftarrow y + V[j]$ 
9:         if  $k > 0 \wedge k \leq n \wedge l > 0 \wedge l \leq n$  then
10:           $P \cup [k, l]$ 
11:        end if
12:      end if
13:    end for
14:  end for
15:  return  $P$ 
16: end procedure
```

---

---

**Algorithm 2** Verificación jugadas

---

```
1: procedure TODOVISITADO( $T, n$ )
2:    $v \leftarrow True$ 
3:   for  $i \leftarrow 1$  to  $n$  do
4:     for  $j \leftarrow 1$  to  $n$  do
5:       if  $T[i][j] = 0$  then
6:          $v \leftarrow False$ 
7:       end if
8:     end for
9:   end for
10:  return  $v$ 
11: end procedure
```

---

---

**Algorithm 3** Encontrar camino

---

```
1: procedure CABALLO( $T, x, y, n$ )
2:    $T[x][y] \leftarrow 1$ 
3:    $P \leftarrow CalcularPosiciones(x, y, n)$ 
4:    $sol \leftarrow TodoVisitado(T, n)$ 
5:    $i \leftarrow 1$ 
6:   while  $sol = False \wedge i \leq |P|$  do
7:      $act \leftarrow P[i]$ 
8:     if  $T[act.x][act.y] \neq 1$  then
9:        $T[act.x][act.y] \leftarrow 1$ 
10:       $\langle sol, S \rangle \leftarrow Caballo(T, act.x, act.y, n)$ 
11:      if  $sol = True$  then
12:         $p \leftarrow [act.x, act.y]$ 
13:         $S \cup p$ 
14:      else
15:         $T[act.x][act.y] \leftarrow 0$ 
16:        Clear  $S$ 
17:      end if
18:    end if
19:     $i \leftarrow i + 1$ 
20:  end while
21:  return  $\langle sol, S \rangle$ 
22: end procedure
```

---

---

**Algorithm 4** Encontrar camino auxiliar

---

```
1: procedure FINDPATH( $n$ )
2:   let  $T \leftarrow \mathbb{N}^{n \times n}$ 
3:   let  $S$  be an empty list
4:    $path \leftarrow False$ 
5:    $i, j \leftarrow 0, 0$ 
6:   while  $path = False \wedge i < n$  do
7:     while  $path = False \wedge j < n$  do
8:        $path, S \leftarrow Caballo(T, i, j, n)$ 
9:        $j \leftarrow j + 1$ 
10:    end while
11:     $i \leftarrow i + 1$ 
12:  end while
13:  return  $\langle path, S \rangle$ 
14: end procedure
```

---

#### 4. Análisis de la Solución

La solución presentada encuentra la secuencia que debe dar el caballo para recorrer todo el tablero. Dada una posición inicial, calcula las posibles casillas a las cuales puede saltar la ficha y luego determina si por ese camino se llega a una solución, este proceso se repite hasta que se encuentra un camino o se hallan recorrido todas las posiciones y con ninguna de ellas se pudo llegar a la solución.

#### 5. Invariante

La secuencia  $S$  contiene los saltos correctos que el caballo ha dado hasta cierto momento y el tablero  $T$  que contiene las posiciones visitadas y no visitadas hasta el momento.

#### 6. Complejidad

Para la función CalcularPosiciones por inspección se determina que su complejidad es  $O(1)$ , ya que siempre calcula los ocho posibles movimientos que puede hacer el caballo.

Para TodoVisitado la complejidad por inspección es  $O(n^2)$  puesto que recorre todo el tablero verificando que se haya visitado.

Finalmente para la función Caballo por inspección su complejidad es  $O(n!)$ , porque cada vez que el caballo cambia de posición re-calcula las posibles posiciones. En una prueba realizada con un tablero de tamaño  $8 \times 8$  se tardó aproximadamente 30 minutos en hallar la solución.