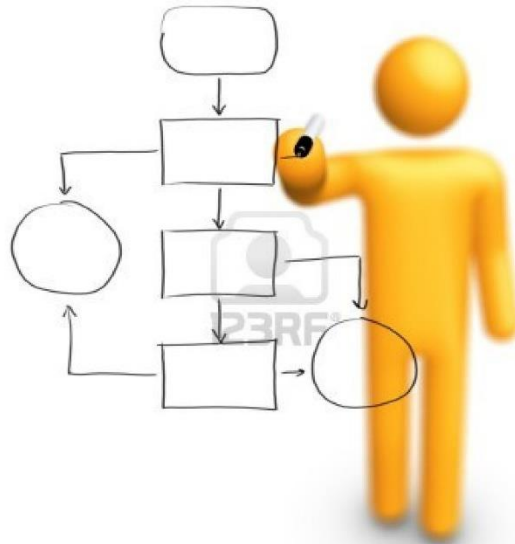




# INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA  
CAMPUS ZACATECAS

## Análisis de Algoritmos



## Reporte de Algoritmo de n-reinas

Docente: Cruz Leija Roberto Oswaldo

Alumno: Ismael Cortés Castillo

3CM1

29 de noviembre de 2019

Zacatecas, Zacatecas

## Introducción

El problema de las  $n$ -Reinas es muy antiguo, propuesto por primera vez en el año 1848, consiste en encontrar una asignación a  $n$  reinas en un tablero de ajedrez de  $n \times n$  de modo tal, que estas no se ataquen. En el presente reporte se presentan las diferentes pruebas para el algoritmo proporcionado por el profesor; las pruebas se realizaron para 8, 15, 30, 70 y 90 reinas, para sus respectivos tableros.

## Descripción

Los datos que más se modificaban (de la clase AlgoritmosGeneticos2019) eran:

- *numGeneraciones*
- *tamPoblacion*
- *probMuta*
- *pMuestra*
- *tamGenotipo*

Estos datos se modificaron de tal manera que el algoritmo resolviera el problema en el menor tiempo posible, y que, en el mejor caso, se obtuviera la mejor solución para el problema.

A continuación, se muestran cada una de las configuraciones que se usaron para resolver el problema para cada una de las pruebas propuestas en clase.

### Para 8 reinas en un tablero de 8x8:

```
Configuracion c1 = new Configuracion(2000000, 64, 0.2, 0.01, new
Seleccion.TipoSeleccion[]{Seleccion.TipoSeleccion.RANDOM, Seleccion.TipoSeleccion.T ORNEO}, 8);
```

Solución:

g: 36 [6, 2, 7, 1, 4, 0, 5, 3]

### Para 15 reinas en un tablero de 15x15

```
Configuracion c1 = new Configuracion(2000000, 225, 0.2, 0.01, new
Seleccion.TipoSeleccion[]{Seleccion.TipoSeleccion.RANDOM, Seleccion.TipoSeleccion.T ORNEO}, 15);
```

Solución:

g: 10418 [12, 8, 4, 0, 14, 9, 1, 3, 7, 10, 13, 2, 5, 11, 6]

### Para 30 reinas en un tablero de 30x30

```
Configuracion c1 = new Configuracion(10000, 900, 0.2, 0.001, new  
Seleccion.TipoSeleccion[]{Seleccion.TipoSeleccion.RANDOM,Seleccion.TipoSelecc  
cion.T ORNEO}, 30);
```

Solución:

g: 2237 [14, 19, 21, 1, 26, 20, 7, 4, 13, 17, 0, 27, 22, 6, 25, 28, 16, 5, 10, 8, 11, 24, 2, 29, 18, 12, 3, 9, 23, 15]

### Para 70 reinas en un tablero de 70x70

```
Configuracion c1 = new Configuracion(10000, 60, 0.3, 0.001, new  
Seleccion.TipoSeleccion[]{Seleccion.TipoSeleccion.RANDOM,Seleccion.TipoSelecc  
cion.T ORNEO}, 70);
```

Solución:

g: 1732 [40, 35, 10, 12, 20, 29, 37, 19, 62, 60, 56, 39, 13, 49, 21, 7, 3, 0, 14, 44, 9, 59, 16, 64, 23, 58, 47, 17, 5, 32, 45, 61, 27, 43, 51, 2, 68, 8, 18, 50, 66, 38, 48, 15, 63, 26, 31, 67, 52, 54, 1, 4, 36, 11, 30, 41, 69, 55, 28, 22, 42, 34, 6, 65, 57, 53, 25, 46, 33, 24]

### Para 90 reinas en un tablero de 90x90

```
Configuracion c1 = new Configuracion(10000, 60, 0.3, 0.001, new  
Seleccion.TipoSeleccion[]{Seleccion.TipoSeleccion.RANDOM,Seleccion.TipoSelecc  
cion.T ORNEO}, 90);
```

Solución:

g: 3652 [16, 11, 77, 32, 19, 12, 23, 21, 60, 64, 56, 75, 24, 89, 40, 17, 22, 42, 31, 79, 71, 83, 4, 58, 69, 63, 26, 35, 15, 48, 88, 61, 73, 28, 78, 84, 59, 34, 27, 57, 38, 9, 66, 1, 45, 76, 82, 67, 36, 6, 33, 72, 8, 86, 13, 87, 20, 7, 43, 51, 37, 39, 10, 85, 30, 55, 65, 46, 14, 53, 81, 3, 68, 44, 41, 0, 80, 49, 2, 54, 50, 18, 47, 74, 70, 5, 25, 29, 62, 52]

## Conclusión

La probabilidad de mutación influye en el proceso en el que se busca la mejor solución, ya que si no lo hiciera el programa se estancaría, aunque esto en algunas ocasiones lo llega a empeorar e incluso no se llega a encontrar solución a la configuración propuesta