

Documentación Práctica 8
Estructura de Datos Lineales
Recursividad y Backtracking
Problema de las N Reinas

Introducción.

Como se explicó en la práctica anterior la recursividad es resolver un problema grande dividiéndolo en sub-problemas hasta llegar a un caso base para resolver el problema original.

En esta práctica también se hará uso del “backtracking” que nos indicará si una solución es correcta y si no lo es se regresa a su estado previo y continua a la siguiente posible solución, esto hasta encontrar algunas o todas las posibles soluciones a un problema.

Descripción del Problema.

Se tiene un tablero de ajedrez de $N \times N$ con N número de reinas. Las reinas se deberán colocar en el tablero de tal manera que ninguna reina se ataque entre sí, hay que considerar que una reina en el ajedrez se puede mover de manera horizontal, vertical, y diagonalmente.

Se verifica que no exista reina colocada en la columna y que en la casilla que se quiera colocar la reina no sea atacada por las diagonales y si no se puede colocar se hace backtracking y se avanza a la siguiente posición del renglón. El algoritmo finalizará cuando se encuentren todas las soluciones posibles.

Desarrollo.

A continuación, se mostrarán los pseudocódigos de los métodos que se implementarán para resolver el problema. Es importante mencionar que N es una constante cuyo valor puede ser modificado si se requiere.

Pseudocódigo Imprime Tablero.

- Entrada: Arreglo de N por N entero: tablero.
Salida: Tablero

```
Función Vacío ImprimeTablero(tablero)
    cont <- 1
    Llama Escribe (“Solución: “ , cont)
    Para i <- 0 hasta N hacer
        Para j <- 0 hasta N hacer
            Llama Escribe (tablero[i][j])
        Fin Para
    Fin Para
    cont <- cont + 1
Fin Función
```

Pseudocódigo No Ataque.

- Entrada: Arreglo de N por N entero: tablero, entero ren, entero col
Salida: Verdadero o Falso

```
Función Booleano noAtaque(tablero, ren, col)
  Para i <- 0 Hasta N Hacer
    Si (tablero[ren][i]) entonces
      Retorna Falso
    Fin Si
  Fin Para
  Para i <- ren, j <- col Hasta 0 Hacer // tanto i como j disminuyen
    Si (tablero[i][j]) entonces
      Retorna Falso
    Fin Si
  Fin Para
  Para i <- ren, j <- col Hasta j > 0 e i < N Hacer // i aumenta, j disminuye.
    Si (tablero[i][j]) entonces
      Retorna Falso
    Fin Si
  Fin Para
Fin Función
```

Pseudocódigo Coloca Reina.

- Entrada: Arreglo de N por N entero: tablero, entero col
Salida: Falso o Verdadero

```
Función Booleano colocaReina(tablero, col)
  Si (col = N) entonces
    Llamada a Función ImprimeTablero(tablero)
    Retorna Verdadero
  Fin Si
  Para i <- 0 hasta N hacer
    Si (Llamada a función noAtaque, i, col) entonces
      tablero[i][col] <- 1;
      colocaReina(tablero, col + 1)
      tablero[i][col] <- 0;
    Fin Si
  Fin Para
  Retorna Falso
Fin Función
```

Pseudocódigo Tablero Reinas.

- Entrada: Arreglo de N por N entero: tablero
Salida: Tablero o Llama Escribe ("La solución no Existe.")

```
Función Vacío tableroReinas()  
  Para ren <- 0 Hasta N Hacer  
    Para col <- 0 Hasta N Hacer  
      tablero[ren][col] <- 0  
    Fin Para  
  Fin Para  
  Si (Llamada a Función colocaReina(tablero, 0))  
    Llama Escribe ("La solución no Existe.")  
  Regresa  
Fin Si  
Fin Función
```

Resultados.

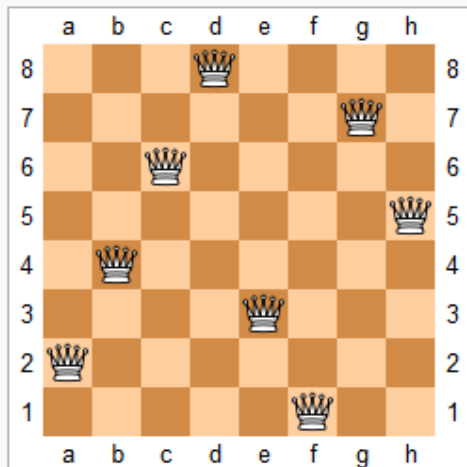
Usando los métodos anteriores es posible conocer todas las posibles soluciones en las que se pueden colocar N reinas en un tablero de N por N. En esta práctica se comprobaron los resultados con un tablero de 5 x 5 con 5 reinas, tablero de 6 x 6 con 6 reinas, tablero de 8 x 8 con 8 reinas y tablero de 17 x 17 con 17 reinas.

Los resultados fueron los siguientes:

Tablero de 5 x5 con 5 Reinas	10 soluciones
Tablero de 6 x 6 con 6 Reinas	4 soluciones
Tablero de 8 x 8 con 8 Reinas	92 soluciones
Tablero de 14 por 14 con 14 Reinas	365596 soluciones

Es importante mencionar que estas no son solo soluciones únicas al problema ya que muchas de estas se pueden obtener de la traslación, rotación y simetría de las soluciones únicas, en un tablero de 8 x 8 existen 12 soluciones únicas.

Una solución del problema de las 8 reinas es el siguiente:



En la consola de LINUX se mostrará la solución del problema de ésta manera:

Solución: 1.

```
0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0
0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0
```

Manual de Usuario:

- Para compilar el programa es necesario escribir lo siguiente en la terminal de LINUX:
 - `g++ /home/nombreUsuario/ubicaciónArchivo/nombreArchivo.cpp -o p`
 - Siendo “nombreArchivo” el nombre que lleva el programa, en este caso lleva por nombre: problemaReinas.cpp
- Para ejecutar el programa se introduce lo siguiente:
 - `./p`
 - La última letra de la línea de compilación indica la letra que va después del “./”.
- Si se desea cambiar el número de reinas se modifica la línea 5 del código:
 - `const int N = número de reinas;`
 - Ejemplo: `const int N = 12;`