



- **Tecnológico Nacional de México campus Culiacán**
- **Tecnologías de la Información y las Comunicaciones**
- **Culiacan, Sinaloa**
- **09 de Marzo del 2025**
- **Rafael Patron Marco Antonio**

PROBLEMA DE LAS 8 REYNAS

El problema de las 8 reinas consiste en colocar 8 reinas en un tablero de ajedrez de 8x8 sin que se ataquen entre sí. Una reina puede atacar en horizontal, vertical y diagonal, por lo que ninguna reina debe compartir la misma fila, columna o diagonal.

Esto se realizara utilizando el método de búsqueda Tabú que conlleva a ser:

Optimización matemática que se utiliza para resolver problemas.

Es una técnica de búsqueda local que mejora su rendimiento al evitar repetir soluciones ya visitadas.

Inicio

El proceso comienza con la definición del tamaño del tablero (N) y la generación de una solución inicial aleatoria. Esto implica colocar una reina en cada columna del tablero al azar. Posteriormente, se evalúa esta solución inicial contando el número de conflictos entre las reinas, es decir, el número de pares de reinas que se amenazan mutuamente.

Búsqueda del Vecindario

Se genera el vecindario de la solución actual moviendo cada reina a cada una de las otras filas en su columna, creando así soluciones vecinas. Cada una de estas soluciones vecinas se evalúa contando el número de conflictos entre las reinas.

Selección del Vecino No Tabú

Se identifica el mejor vecino, es decir, aquel con el menor número de conflictos. Se verifica si este vecino está en la lista tabú. Si lo está, se aplica el criterio de aspiración: si el vecino tabú es mejor que la mejor solución encontrada hasta ahora, se acepta.

Criterio de Terminación

Se verifica si se ha encontrado una solución sin conflictos (número de conflictos = 0) o si se ha alcanzado un número máximo de iteraciones. Si se cumple alguno de estos criterios, se pasa al paso de finalización; de lo contrario, se vuelve al paso de búsqueda del vecindario.

```
def free(row, col, n):  
    """ Determina si la casilla rowxcol está libre de ataques.  
  
    @param row: Fila  
    @param col: Columna  
    @return: True si la casilla está libre de ataques por otras reinas.  
    """  
  
    for i in range(n):  
        if tablero[row][i] == 'R' or tablero[i][col] == 'R':  
            return False  
  
    if row <= col:  
        c = col - row  
        r = 0  
    else:  
        r = row - col  
        c = 0  
    while c < n and r < n:  
        if tablero[r][c] == 'R':  
            return False  
        r += 1  
        c += 1  
  
    if row <= col:  
        r = 0  
        c = col + row  
        if c > n-1:  
            r = c - (n-1)  
            c = n-1  
    else:  
        c = n-1  
        r = row - ((n-1) - col)  
  
    while c >= 0 and r < n:  
        if tablero[r][c] == 'R':  
            return False  
        r += 1  
        c -= 1
```

```

    return True

def agregar_reina(n,m):
    """ Agrega n reinas al tablero.

    @param: n El número de reinas a agregar
    @return True si se pudo agregar las reinas requeridas
    """

    if m < 1:
        return True

    for idx_row in range(n):
        for idx_col in range(n):
            if free(idx_row, idx_col, n):
                tablero[idx_row][idx_col] = 'R'
                if agregar_reina(n,m-1):
                    return True
            else:
                tablero[idx_row][idx_col] = 'O'

    return False

n= int(input("DIGITE NUMERO DE REINAS: "))
m=n
tablero = []
for i in range(n):
    tablero.append(['O'] * n)
agregar_reina(n,m)
for row in tablero:
    print(*row)

```

```

❏ DIGITE NUMERO DE REINAS: 10
R O O O O O O O O O
O O R O O O O O O O
O O O O O R O O O O
O O O O O O O R O O
O O O O O O O O R O
O O O O R O O O O O
O O O O O O O O R O

```

CODIGO TOMADO DE: <https://es.stackoverflow.com/questions/171895/solucion-reto-de-las-8-reinas-en-python>