knightstour

November 24, 2022

1 Algoritmo del Tour del Caballo

- Faustino Maggioni Duffy
- Gonzalo Guerrero

Defino el tamaño del tablero

[ ]:
```
n = 8
```

Funcion para chequear si los indices son validos

[ ]:
```
def isSafe(x, y, board):
    if(x >= 0 and y >= 0 and x < n and y < n and board[x][y] == -1):
        return True
    return False
```

Funcion para imprimir el tablero con los pasos resultantes del recorrido del caballo

[ ]:
```
def printSolution(n, board):
    for i in range(n):
        for j in range(n):
            print(board[i][j], end=' ')
        print()
```

solveKT * Esta funcion resuelve el Problema del Tour del Caballo usando backtracking con una funcion auxiliar, solveKTUtil(). Resuelve falso si no hay un camino posible, y verdadero si es que hay. Imprime la solucion y termina. * Es importante aclarar que muestra UNA sola de las posibles soluciones, puede haber mas de una.

[ ]:
```
def solveKT(n):
    board = [[-1 for i in range(n)]for i in range(n)]
    move_x = [2, 1, -1, -2, -2, -1, 1, 2]
    move_y = [1, 2, 2, 1, -1, -2, -2, -1]

    board[0][0] = "00"

    pos = 1

    if(not solveKTUtil(n, board, 0, 0, move_x, move_y, pos)):
```

```
        print("Solution does not exist")
    else:
        printSolution(n, board)
def solveKTUtil(n, board, curr_x, curr_y, move_x, move_y, pos):
    '''
        A recursive utility function to solve Knight Tour
        problem
    '''

    if(pos == n**2):
        return True
    # Try all next moves from the current coordinate x, y
    for i in range(8):
        new_x = curr_x + move_x[i]
        new_y = curr_y + move_y[i]
        if(isSafe(new_x, new_y, board)):
            pos_aux = str(pos)
            if (pos <= 9):
              pos_aux = "0" + pos_aux
            board[new_x][new_y] = pos_aux
            if(solveKTUtil(n, board, new_x, new_y, move_x, move_y, pos+1)):
                return True

            # Backtracking
            board[new_x][new_y] = -1
    return False


# Driver Code
if __name__ == "__main__":

    # Function Call
    solveKT(n)
```

```
00 59 38 33 30 17 08 63
37 34 31 60 09 62 29 16
58 01 36 39 32 27 18 07
35 48 41 26 61 10 15 28
42 57 02 49 40 23 06 19
47 50 45 54 25 20 11 14
56 43 52 03 22 13 24 05
51 46 55 44 53 04 21 12
```