Instituto Politécnico Nacional
Unidad Profesional Interdisiplinaria de Ingenierias campus Zacatecas

Analicis de Álgoritmos

Problema del caballo

Olando Odiseo Belmonte Flores

Maestro:
Roberto Oswaldo Cruz Leija

October 28, 2019

1 Planteamiento del problema

El problema del caballo es un antiguo problema matemáticoen el que se pide que, teniendo un tablero de $n * n$ dimensiones, un caballo de ajedrez colocado en cualquier posición (x, y) sea capaz de recorrer todo el tablero pasando por cada casilla una sola vez, ésto da como resulado un total de $n^2 - 1$ movimietos totales.

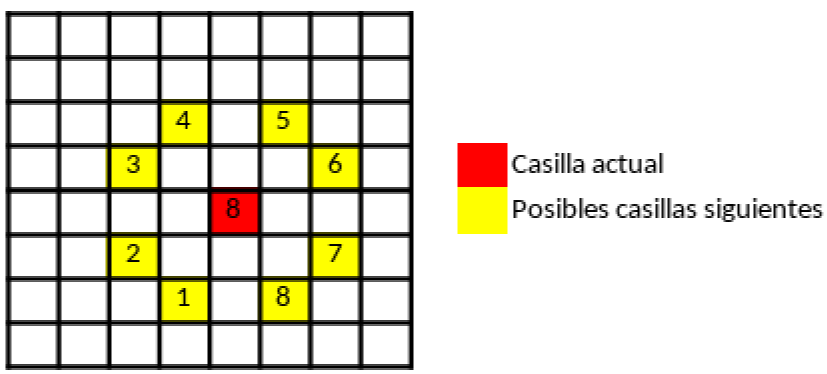
Éste problema matemático ha pasado a ser también, en tiempos actuales, un problema computacional, con el objetivo de que una computadora sea capaz de encontrar el camino con las mismas restrixiones del problema original.

Hay que considerar que el caballo no puede salir del tablero ni brincar de una orilla a la otra, por lo que hay que tomar en cuenta también que tan cerca del borde del tablero nos encontramos.

2 Plantemaiento de la solución

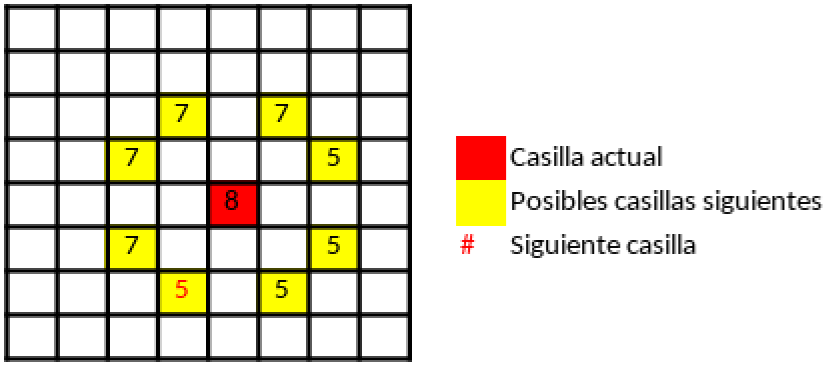
Para la resolución de éste problema se hara uso de un algoritmo avido o voraz, e decir, un algoritmo basado en una heurística a seguir.

Empezemos definiendo el modo de recorrer la busqueda de casillas, en nuestro caso la primer casilla a comprobar estará dos casillas abajo y una a la izquierda de la casilla actual, desde ahí se recorrera en sentido horario como se muestra en la figura.



Una vez establecido el orden de recorrido estableceremos el criterio de selección. Con forme el caballo va recorriendo las casillas del tablero, se van reduciendo los movimientos disponibles por lo que es necesario establecer un criterio que nos permita decidir cuál es el siguiente movimiento.

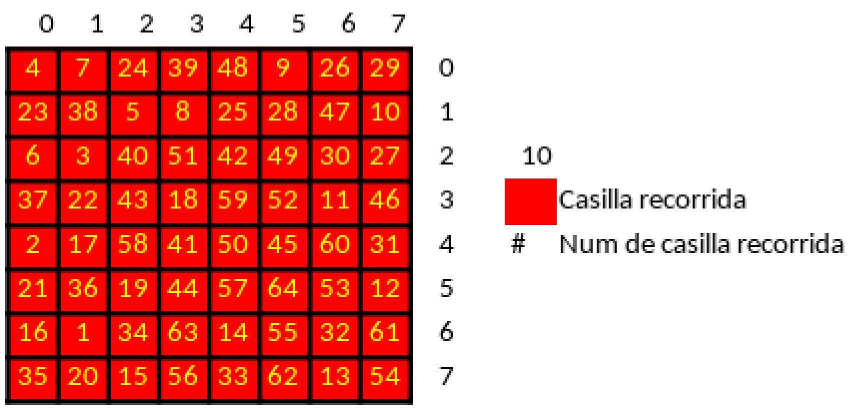
Nos basaremos en el número de casillas que quedan porrecorrer, concretamente el de menor número. Una vez vistos los posibles movimienos que tenemos contaremos cuantas casillas están dsponibles desde cada una de ellas, es decir, si nos colocamos en una de esas casillas, ¿cuáles son los posible smovientos que hay?, como se ha dicho antes se elie el que tenga el menor número. En el caso de que más de una casilla tengan el mismo numero se elije la primera encontrada.



En la imágen anterior el numero dentro de cada casilla indica el numro de movimientos disponibles que hay desde la posición correspondiente, teniendo en cueta que la casilla en rojo (la actual) ya no es posible de recorrer.

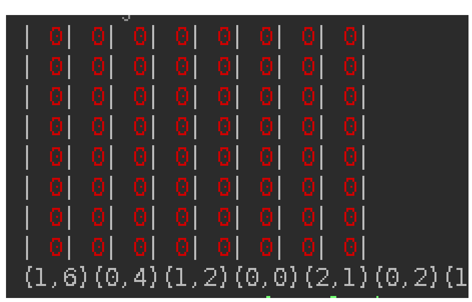
3 Resultados del algoritmo

Antes de aplicar el algoritmo se probó la heirística en una hoja de calculo iniciando desde la posocion $(1, 6)$ y colocando en cada casilla el numero consecutivo del recorrido encontrado.



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

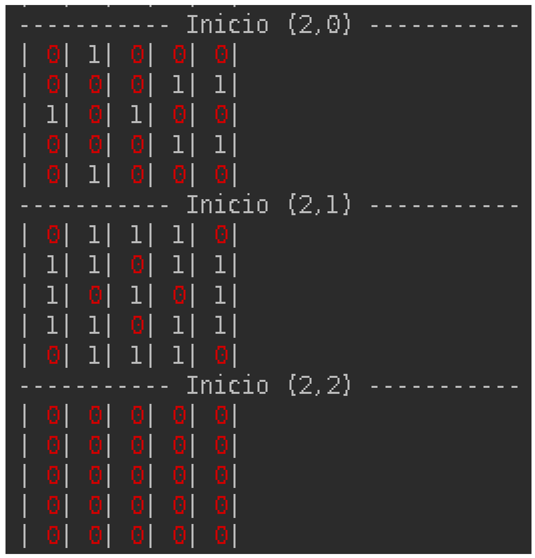
Una vez probada la heurística y realizado el código se hizo la prueba iniciano desde el mismo punto, teniendo el siguiente resultado:



El camino recorrio completo que se muestra debajo del tblero es el siguiente:
(1,6)(0,4)(1,2)(0,0)(2,1)(0,2)(1,0)(3,1)(5,0)(7,1)(6,3)(7,5)(6,7)(4,6)(2,7)(0,6)(1,4)(3,3)(2,5)
(1,7)(0,5)(1,3)(0,1)(2,0)(4,1)(6,0)(7,2)(5,1)(7,0)(6,2)(7,4)(6,6)(4,7)(2,6)(0,7)(1,5)(0,3)(1,1)

(3,0)(2,2)(3,4)(4,2)(2,3)(3,5)(5,4)(7,3)(6,1)(4,0)(5,2)(4,4)(3,2)(5,3)(6,5)(7,7)(5,6)(3,7)(4,5)(2,4)(4,3)(6,4)(7,6)(5,7)(3,6)(5,5)

El output del programa se muestra en 0 rojos para una mejor visualización del resultao, pues esta heurística no funiona para todos los tablaros de diferente tamaño ni para todos los puntos de inicio. Durante las pruebas del algoritmo se observo que para los tableros de tamaño n impar existen algunos puntos de inicio para los cuales no hay solucion, los dos siguientes ejemplos son para un tablero de $n = 5$ en las posiciones iniciales mostradas arriba de cada tablero.



Como se puede observar para las posiciones iniciales $(2, 0)$ Y $(2, 1)$ no existe una solución, sin embargo para la posición $(2, 2)$ si la hay.

Nota: en los tableros, cada casilla marcada con '1' o ha sido recorrida; de igual forma las casillas marcdas con '0' ya han sido recorridas.

4 Concluciones

Los algoritmos avidos pueden ser relativamete faciles de diseñar e implementar, pero eso no significa que sea la mejor opción. Un algoritmo avido puede tener soluciones para algunos inputs pero no necesariamente se cumplira para todos los casos. Eso, aunado a que si el problema tiene un grado de complejidad alto variante puede generar una gran carga al procesador.

Por ejemplo, en el problema del caballo segun aumente el tamaño de n, el tiempo en el que encuentra la solución aumentara de forma cuadratica, si además queremos que encuentre la solución para cada punto de inicio su complejidad aumentara de manera exponencial, pues existen $n!$ puntos de inicio para los cuales hay, como mucho, $n^2 - 1$ pasos a recorrer en cada uno de ellos.