Instituto Politécnico Nacional

Programa Académico: Ingeniería en Sistemas Computacionales
Unidad de Aprendizaje: Análisis de Algoritmos
Alumno: Nava Álvarez José Andrés
Fecha: 28/10/2019

Problema de la Marcha del Caballo

Tabla de contenido

Introducción

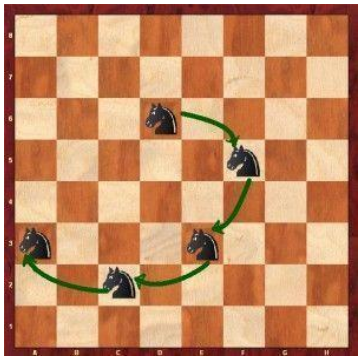
En la presente actividad realizamos el análisis de un antiguo problema matemático, el mítico "Problema del Caballo" en el cual pusimos a prueba lo aprendido en cuanto al análisis de algoritmos. Posteriormente llevamos a cabo la realización de un programa para resolver dicho problema, apoyados por supuesto en la programación dinámica.

Marco Teórico

Problema del Caballo

Este problema está basado en la pieza del caballo que se utiliza en el ajedrez. El movimiento de esta pieza es diferente al de las demás puesto que no es lineal. El movimiento del caballo es peculiar ya que estando en una casilla negra solo podrá avanzar a una blanca y viceversa.

El movimiento del caballo se da en forma de L y normalmente tendrá hasta 8 posiciones posibles, aunque este número puede reducirse a 2 si se encuentra en las orillas del tablero.



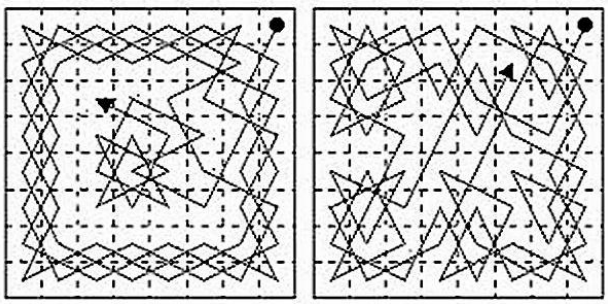
Movimiento del Caballo

Entre los problemas matemáticos inspirados en el ajedrez, uno de los más interesantes es el problema de la marcha del caballo. Consiste en recorrer las 64 casillas del tablero con un caballo, en 64 movimientos y sin pasar dos veces por la misma casilla.

Existen dos opciones:

- Empezar y terminar en la misma casilla (circuito cerrado).
- Empezar en una casilla y terminar en otra (circuito abierto).

Para este caso usaremos cualquiera de los dos caminos.



Camino Abierto **Camino Cerrado**

Programacion Dinamica

En informática, la programación dinámica es un método para reducir el tiempo de ejecución de un algoritmo mediante la utilización de subproblemas superpuestos y subestructuras óptimas.

Una subestructura óptima significa que se pueden usar soluciones óptimas de subproblemas para encontrar la solución óptima del problema en su conjunto. Por ejemplo, el camino más corto entre dos vértices de un Grafo se puede encontrar calculando primero el camino más corto al objetivo desde todos los vértices adyacentes al de partida, y después usando estas soluciones para elegir el mejor camino de todos ellos. En general, se pueden resolver problemas con subestructuras óptimas siguiendo estos tres pasos:

- Dividir el problema en subproblemas más pequeños.
- Resolver estos problemas de manera óptima usando este proceso de tres pasos recursivamente.
- Usar estas soluciones óptimas para construir una solución óptima al problema original.

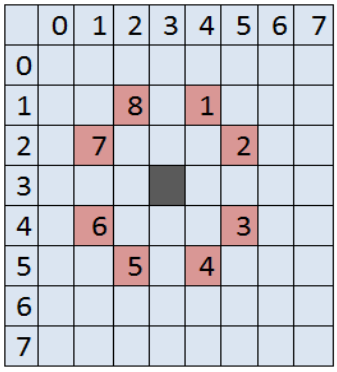
Los subproblemas se resuelven a su vez dividiéndolos en subproblemas más pequeños hasta que se alcance el caso fácil, donde la solución al problema es trivial.

Desarrollo

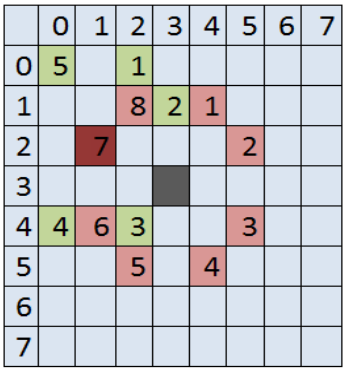
Empezaremos analizando el problema y la manera de abordarlo. Primero necesitamos conocer las diferencias entre el problema del caballo tradicional y el que vamos a utilizar; En nuestro caso no utilizaremos un tablero de ajedrez de 8x8 casillas, sino, un tablero de dimensiones cambiantes, Aunque siempre de forma cuadrada.

- Algunas otras consideraciones para su programación son:
- Tomar en cuenta que a excepción del primer movimiento el caballo tendrá siete o menos posibles movimientos posibles.
- Al momento de calcular los posibles movimientos debemos tomar en cuenta que no se salgan del tablero.
- Tomando en cuenta un tablero de dimensiones nxn el numero de saltos será (nxn)-1.
- En este caso al momento de elegir el siguiente movimiento favoreceremos a los que tengan menos subsecuentes movimientos posibles.

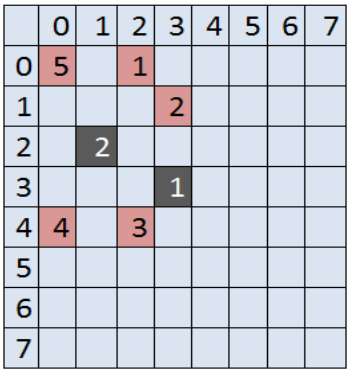
Teniendo en cuenta lo anterior, nuestro algoritmo comenzara con calcular las posiciones posibles desde el inicio.



Despues de localizar los posibles movimiento calcularemos ahora los posibles movimientos de todos estos puntos y eligiremos el que tenga menos movimientos subsecuentes posibles



Una vez encontrado el movimiento con los menores subsecuentes posibles, avanzaremos a esa casilla y comenzaremos el proceso de nuevo. Esto hasta recorrer todas las casillas o hasta quedarnos sin posibles movimientos.



Al momento de implementar el programa haremos que pruebe con todas las casillas como inicio para encontrar en cuales si es posible realizar el camino y en cuáles no lo es.

Resultados

A continuación se muestran los resultados de diversas pruebas.

- Pruebas con un tablero de 5x5;

```
Inicio 0 , 0
[1] [12] [25] [18] [7]
[22] [17] [8] [13] [24]
[11] [2] [23] [6] [19]
[16] [21] [4] [9] [14]
[3] [10] [15] [20] [5]
```

Primera prueba exitosa

```
Inicio 0 , 1
No tiene Solucion
[0] [1] [6] [23] [12]
[7] [24] [11] [18] [5]
[2] [17] [22] [13] [10]
[21] [8] [15] [4] [19]
[16] [3] [20] [9] [14]
```

Primera prueba fallida

- Pruebas con un tablero de 8x8;

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

Prueba exitosa (Todas tienen solución)

- Pruebas con un tablero de 4x4;

```
Inicio 3 , 3
No tiene Solucion
[3] [14] [7] [0]
[8] [11] [4] [15]
[13] [2] [9] [6]
[10] [5] [12] [1]
```

Ultima prueba fallida (Ninguna tiene solución)

Conclusiones

Con los resultados obtenidos en las pruebas con el software podemos concluir ciertas propiedades del problema del caballo. Por supuesto tomando en cuenta que el problema fue resuelto y atacado en base a seguir el camino con el menor número de posibles caminos.

- El tablero más pequeño del cual se pueden sacar soluciones es el de 5x5 ya que de dimensiones inferiores no se obtiene ninguna solución.
- Para tableros de dimensión par mayor a cuatro se obtienen soluciones en cualquier casilla que utilicemos como inicio.
- Para tableros de dimensión impar mayor a 3 obtendremos soluciones en algunas de su casilla mientras que en otra no lo habrá.