Universidad Tecnológica Metropolitana

Facultad de Ingeniería

Departamento de Informática y Computación

Análisis de Algoritmos INF-648

**“Algoritmo de las 8 reinas”**

**Integrantes**: Aldo Aguilar

Gerson Leiva

Camilo Torres

Sebastián Maureira

**Académico**: Alejandro Reyes

**Ayudante**: Paola Contreras

**Introducción**

El problema fue originalmente propuesto en 1848 por el ajedrecista Max Bezzel, y durante los años, muchos matemáticos, incluyendo a Gauss y a Georg Cantor, han trabajado en este problema y lo han generalizado a n-reinas. Las primeras soluciones fueron ofrecidas por Franz Nauck en 1850. Nauck también se abocó a las n-reinas (en un tablero de nxn de tamaño arbitrario). En 1874, S. Günther propuso un método para hallar las soluciones usando determinantes, y J.W.L. Glaisher redefinió su aproximación.

**Problemática**

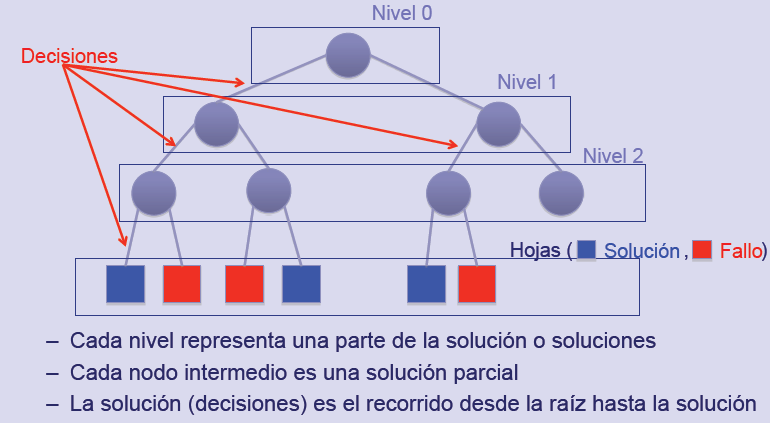
El problema consiste en colocar en un tablero de ajedrez (8×8) ocho reinas sin que puedan atacarse entre ellas. El método de resolución es ir colocando una reina por vez e ir analizando que lugares quedan libres para el resto de las reinas.

**Metodología Backtraking**

Una de las metodologías aplicadas en este algoritmo de 8 reinas es el backtraking. Técnica para recorrer sistemáticamente todas las posibles configuraciones de un espacio.

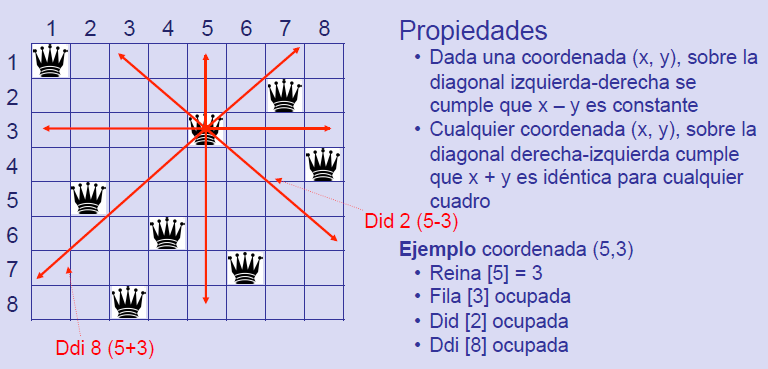
En su forma básica la Vuelta Atrás se asemeja a un recorrido en profundidad dentro de un árbol cuya existencia sólo es implícita, y que se denomina árbol de expansión. Este árbol es conceptual, en donde cada nodo de nivel k representa una parte de la solución y está formado por k etapas que se suponen ya realizadas. Sus hijos son las prolongaciones posibles al añadir una nueva etapa.

Para examinar el conjunto de posibles soluciones es suficiente recorrer este árbol construyendo soluciones parciales a medida que se avanza en el recorrido. En este recorrido pueden suceder dos cosas. La primera es que tenga éxito si, procediendo de esta manera, se llega a una solución (una hoja del árbol). Si lo único que buscábamos era una solución al problema, el algoritmo finaliza aquí; ahora bien, si lo que se busca eran todas las soluciones o la mejor de entre todas ellas, el algoritmo seguirá explorando el árbol en búsqueda de soluciones alternativas (para este caso, serían las casillas en el tablero de ajedrez, es decir, en la matriz). Por otra parte, el recorrido no tiene éxito si en alguna etapa la solución parcial construida hasta el momento no se puede completar; llamamos nodos de fracaso. En tal caso, el algoritmo vuelve atrás (y de ahí su nombre) en su recorrido eliminando los elementos que se hubieran añadido en cada etapa a partir de ese nodo. Si en este retroceso, si existe uno o más caminos aún no explorados que puedan conducir a solución, el recorrido del árbol continúa por ellos.

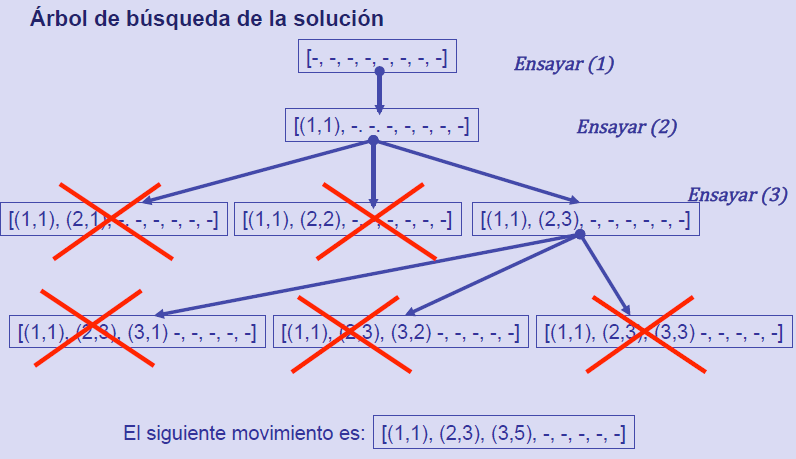


**Metodología backtraking**

**Ejemplo para una situcion de las 8 reinas**



La situación anterior puede ser representada como un árbol de búsqueda de la solución (ver figura Árbol de búsqueda de la solución), donde se van verificando todas las posibles soluciones dentro del tablero en donde no queden en jaque las reinas.



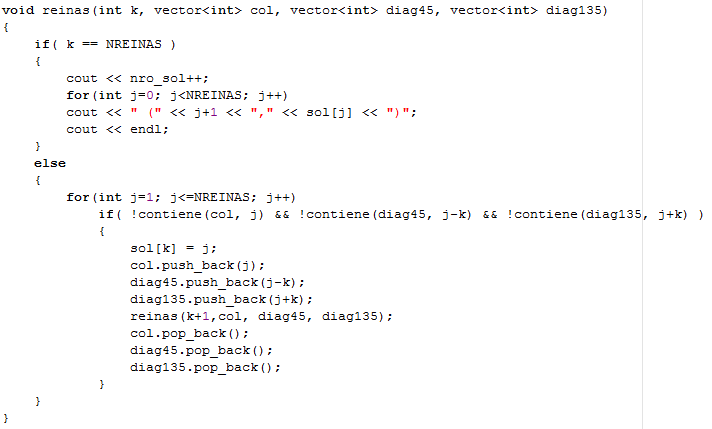
**Árbol de búsqueda para las 8 reinas.**

**Análisis**

La solución a este rompecabezas pasa, por tanto, en situar una reina en cada fila y en cada columna. Preguntándose como se actuará para situar las ocho reinas. Comenzará por la primera columna y situará la primera reina. Al hacer esto, elimina como posibles casillas donde localizar reinas la fila, la columna y las dos diagonales. Teniendo en cuenta estas restricciones, se sitúa en la segunda columna la segunda reina y se "tachan" las nuevas casillas prohibidas. Seguidamente se procede a poner la tercera reina y así sucesivamente. Suponiendo que se han situado las seis primeras reinas. Puede llegar un momento en el que no se pueda situar la séptima sin que ninguna otra le dé jaque. En ese momento, se deshace el movimiento que ha ubicado la sexta reina y busca otra posición válida para dicha reina. Si no se pudiera, se vuelve deshacer el movimiento de la sexta reina y se intenta buscar otra localización. En el caso de que no sea posible, se sigue hacia atrás intentando colocar la reina quinta. Si se puede colocar, entonces se procederá a dejar en el tablero la sexta de nuevo. Si se ha conseguido sin que le den jaque, se intentará emplazar la séptima y, por último, la octava. En general, si hay algún problema, se deshace el último movimiento y se prueba a localizar una casilla alternativa. Si no se consigue, se vuelve hacia atrás y así sucesivamente. En ella se van dando pasos hacia delante mientras sea posible y se deshacen cuando se ha llegado a una situación que no conduce a la resolución del problema original.

**Recursividad para solventar este problema**

Dado que se ha situado una reina en una posición correcta en una columna, hay que considerar el problema de situar otra reina en la columna siguiente: resolver el mismo problema con una columna menos. En este momento realizaría una llamada recursiva (void reinas, en el código del programa). Cuando el tablero se haya reducido a uno de tamaño cero, en cuyo caso no se hará nada. Por tanto, se partirá de un problema de tamaño 8, y se considerarán problemas de tamaño menor quitando una columna en cada llamada recursiva hasta que se llegue a un tablero de tamaño cero, momento en el cual habrá alcanzado el caso base. Si se puede situar una reina en la columna correspondiente, se hace una llamada recursiva, si no, se vuelve hacia atrás y se prueba otras posiciones.



**Función recursiva del programa para resolver el problema de las 8 reinas**

El algoritmo comprueba primero si k=8 , si esto es cierto resulta que tenemos ante nosotros un vector 8-prometedor, lo cual indica que cumple todas las restricciones originando una solución. Si k es distinto de 8, el algoritmo explora las extensiones(k+1)-prometedoras, para ello realiza un bucle, el cual va de 1 a 8, debido al número de reinas. En este bucle se comprueba si entran en jaque las reinas colocadas en el tablero, si no entran en jaque, se realiza una recurrencia en la cual incrementamos k (buscamos(k+1) -prometedor) y añadimos la nueva fila, columna y diagonales al conjunto de restricciones. Al realizar la recurrencia hemos añadido al vector sol una nueva reina la cual no entra en jaque con ninguna de las anteriores, además hemos incrementado el conjunto de restricciones añadiendo una nueva fila, columna y diagonales (una positiva y otra negativa) prohibidas.

**Resultados**

|  |  |  |
| --- | --- | --- |
| Reinas | Soluciones | Tiempo aproximado |
| 1 | 1 | 1 milisegundo |
| 2 | 0 | 0 milisegundos |
| 3 | 0 | 0 milisegundos |
| 4 | 2 | 9 milisegundos |
| 5 | 10 | 77 milisegundos |
| 6 | 4 | 38 milisegundos |
| 7 | 40 | 130 milisegundos |
| 8 | 92 | 287 milisegundos |