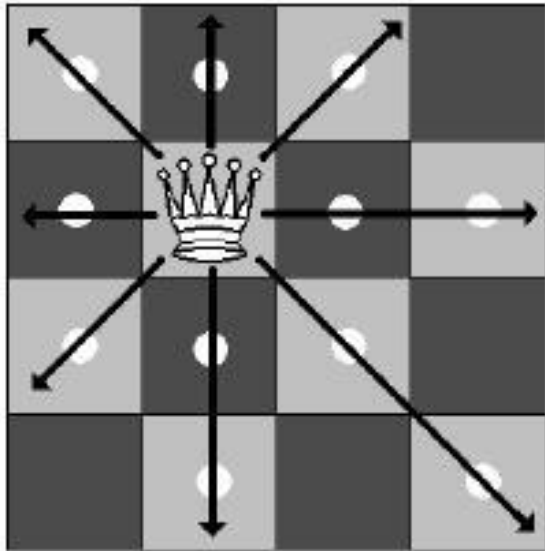


Problema de las ocho reinas



Movimientos posibles de una reina en un tablero de 4x4.

Una posible solución entre las 92 posibles soluciones en un tablero de 8x8

El **problema de las ocho reinas** es un pasatiempo en el que se colocan **ocho reinas** sin que se amenacen. Fue propuesto por el ajedrecista alemán **Max Bezzel** en 1848^[cita requerida]. En el juego del **ajedrez** la reina amenaza a aquellas piezas que se encuentren en su misma fila, columna o diagonal. El juego de las 8 reinas consiste en colocar sobre un tablero de ajedrez ocho reinas sin que estas se amenacen entre ellas. Para resolver este problema emplearemos un esquema **vuelta atrás** (o Backtracking).

1 Historia

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.

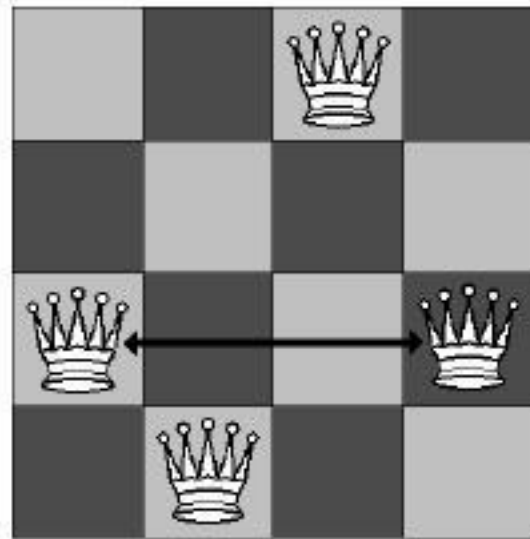
Edsger Dijkstra usó este problema en 1972 para ilustrar el poder de la llamada **programación estructurada**. Él pu-

blicó una descripción altamente detallada del desarrollo del algoritmo de **backtracking**, "depth-first".

Este acertijo apareció en el popular juego de computadora de los '90 llamado "**The 7th Guest**".

2 Planteamiento del Problema

Como cada reina puede amenazar a todas las reinas que estén en la misma fila, cada una ha de situarse en una fila diferente. Podemos representar las 8 reinas mediante un vector[1-8], teniendo en cuenta que cada índice del vector representa una fila y el valor una columna. Así cada reina estaría en la posición $(i, v[i])$ para $i = 1-8$.



Ejemplo de dos reinas amenazadas en el tablero de 4 por 4.

El **vector** (3, 1, 6, 2, 8, 6, 4, 7) significa que la reina 1 está en la columna 3, fila 1; la reina 2 en la columna 1, fila 2; la reina 3 en la columna 6, fila 3; la reina 4 en la columna 2, fila 4; etc... Como se puede apreciar esta solución es incorrecta ya que estarían la reina 3 y la 6 en la misma columna. Por tanto el vector correspondería a una permutación de los ocho primeros **números enteros**.

El problema de las filas y columnas lo tenemos cubierto, ¿pero qué ocurre con las diagonales? Para las posiciones sobre una misma diagonal descendente se cumple que tienen el mismo valor $fila - columna$, mientras que para las posiciones en la misma diagonal ascendente se cumple que tienen el mismo valor $fila + columna$. Así, si

tenemos dos reinas colocadas en posiciones (i, j) y (k, l) entonces están en la misma diagonal si y solo si cumple:

$$i - j = k - l \text{ o } i + j = k + l$$

$$j - l = i - k \text{ o } j - l = k - i$$

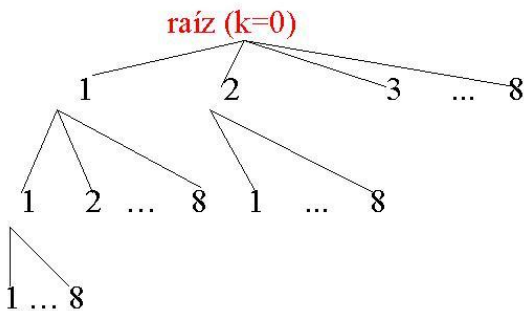
Teniendo en cuenta todas estas consideraciones, podemos aplicar el esquema de retroactivamente para implementar las ocho reinas de una manera realmente eficiente. Para ello, reformulamos el problema como problema de búsqueda en un árbol. Decimos que en un vector $V_{1...k}$ de enteros entre 1 y 8 es k -prometedor, para $0 \leq k \leq 8$, si ninguna de las k reinas colocadas en las posiciones $(1, V_1), (2, V_2), \dots, (k, V_k)$ amenaza a ninguna de las otras. Las soluciones a nuestro problema se corresponden con aquellos vectores que son 8-prometedores.

2.1 Establecimiento del algoritmo

Sea N el conjunto de vectores de k -prometedores, $0 \leq k \leq 8$, sea $G = (N, A)$ el grafo dirigido tal que $(U, V) \in A$ si y solo si existe un entero k , con $0 \leq k \leq 8$ tal que

- U es k -prometedor
- V es $(k+1)$ -prometedor
- $U_i = V_i$ para todo $i \in \{1, \dots, k\}$

Este grafo es un árbol. Su raíz es el vector vacío correspondiente a $k = 0$. sus hojas son o bien soluciones ($k = 8$), o posiciones sin salida ($k < 8$). Las soluciones del problema de las ocho reinas se pueden obtener explorando este árbol. Sin embargo no generamos explícitamente el árbol para explorarlo después. Los nodos se van generando y abandonando en el transcurso de la exploración mediante un recorrido en profundidad.



Esquema reducido del árbol de soluciones.

Hay que decidir si un vector es k -prometedor, sabiendo que es una extensión de un vector $(k-1)$ -prometedor, únicamente necesitamos comprobar la última reina que haya que añadir. Este se puede acelerar si asociamos a

cada nodo promotor el conjunto de columnas, el de diagonales positivas (a 45 grados) y el de diagonales negativas (a 135 grados) controlados por las reinas que ya están puestas.

2.2 Descripción del algoritmo

A continuación se muestra el algoritmo que arroja la solución de nuestro problema, en el cual $sol_{1...8}$ es un vector global. Para imprimir todas las soluciones, la llamada inicial es $reinas(0, \emptyset, \emptyset, \emptyset)$.

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.

2.3 Implementación

A continuación se muestra una posible implementación del anterior algoritmo en C++.

```
#include <iostream> #include <sstream> #include
<cstdio> #include <vector> #include <algorithm>
#define NREINAS 8 // dimensiones del tablero y número
de reinas using namespace std; vector<int> sol; int
nro_sol=1; inline bool contiene(const vector<int>& v,
const int val) { return find(v.begin(), v.end(), val) !=
v.end(); } void reinas(int k, vector<int> col, vector<int>
diag45, vector<int> diag135) { if( k == NREINAS ) {
printf("%3d:", nro_sol++); for(int j=0; j<NREINAS;
j++) cout << " (" << j+1 << ", " << sol[j] << ")";
cout << endl; } else { for(int j=1; j<=NREINAS; j++)
if( !contiene(col, j) && !contiene(diag45, j-k) &&
!contiene(diag135, j+k) ) { sol[k] = j; col.push_back(j);
diag45.push_back(j-k); diag135.push_back(j+k);
reinas(k+1, col, diag45, diag135); col.pop_back();
diag45.pop_back(); diag135.pop_back(); } } int main()
{ cout << "SOLUCIONES AL PROBLEMA DE LAS
" << NREINAS << " REINAS"; sol.resize(NREINAS);
reinas(0, vector<int>(), vector<int>(), vector<int>());
return 0; }
```

3 El problema de las n reinas

El problema de las ocho reinas se puede plantear de modo general como problema de las n reinas. El problema consistiría en colocar n reinas en un tablero de ajedrez de $n \times n$ de tal manera que ninguna de las reinas quede atacando a otra.

Su análisis y solución es isomorfo al de las ocho reinas.

3.1 Número de soluciones

4 Soluciones al problema de las ocho reinas

El problema de las ocho reinas tiene 92 soluciones, de las cuales 12 son **esencialmente distintas**, es decir las 92 soluciones existentes se pueden obtener a partir de simetrías y rotaciones de las 12 soluciones únicas, que se muestran a continuación:

- [N-Queens solvers in many programming languages](#)
- [Find your own solution](#)
- [J Somers N-Queen code](#)
- [Atari BASIC](#)
- [Genetic algorithms](#)
- [Haskell/Java hybrid](#)
- [Java](#)
- [Standard ML](#)
- [Quirkasaurus' 8 Queens Solution](#)
- [LISP solution for N-Queens Problem](#)
- [Javascript solution for 8-Queens Problem](#)
- [Las ocho reinas. Solución](#)
- [8 Reinas en PHP](#)
- [N-Queens solution C++](#)

5 Referencias

- Watkins, John J. (2004). *Across the Board: The Mathematics of Chess Problems*. Princeton: Princeton University Press. ISBN 0-691-11503-6.
- Brassard, Gilles; Bratley, Paul (1997). «Exploración de grafos». *Fundamentos de Algoritmia*. Madrid: PRENTICE HALL. ISBN 84-89660-00-X.

6 Véase también

- [Ajedrez](#)

7 Enlaces externos

- Weisstein, Eric W. «QueensProblem». En Weisstein, Eric W. *MathWorld* (en inglés). Wolfram Research.
- [Solutions to the 8-Queens Problem](#)
- [Walter Koster's N-Queens Page](#)
- [Durango Bill's N-Queens Page](#)
- [On-line Guide to Constraint Programming](#)

7.1 Enlaces a soluciones

- [Solución al problema usando el lenguaje de programación Scheme](#)

8 Texto e imágenes de origen, colaboradores y licencias

8.1 Texto

- **Problema de las ocho reinas** Fuente: https://es.wikipedia.org/wiki/Problema_de_las_ocho_reinas?oldid=83543714 Colaboradores: Sabbut, Jecanre, Kordas, RobotQuistnix, Yrbot, DerkeNuke, BOTijo, Tomatejc, Tamorlan, Juanjo Bazan, Kn, CEM-bot, Bezas-eswiki, Teutonium-eswiki, JAnDbot, Muro de Aguas, TXiKiBoT, HiTe, NaSz, VolkovBot, Mytwm, Muro Bot, Edmenb, YonaBot, SieBot, PaintBot, Loveless, Cuchipa, Kuu6, DorganBot, HUB, LuisArmandoRasteletti, Alexbot, Juan Mayordomo, Louperibot, Luckas-bot, Amirobot, Billinghurst, ArthurBot, Xqbot, FrescoBot, Alejandroadán, Vlarha, RedBot, Æneas, Upiicsa, Hackkpo, Eleazan, KLBot2, MetroBot, Elvisor, Rotlink, Ineditable, Ncomputersorg, Zhirose y Anónimos: 29

8.2 Imágenes

- **Archivo:Chess_d45.svg** Fuente: https://upload.wikimedia.org/wikipedia/commons/3/37/Chess_d45.svg Licencia: GFDL Colaboradores: Este gráfico vectorial fue creado con Inkscape. Artista original: en:User:Cburnett
- **Archivo:Chess_l45.svg** Fuente: https://upload.wikimedia.org/wikipedia/commons/c/cd/Chess_l45.svg Licencia: GFDL Colaboradores: Trabajo propio Artista original: en:User:Cburnett
- **Archivo:Chess_qld45.svg** Fuente: https://upload.wikimedia.org/wikipedia/commons/7/7d/Chess_qld45.svg Licencia: CC-BY-SA-3.0 Colaboradores: Este gráfico vectorial fue creado con Inkscape. Artista original: en:User:Cburnett
- **Archivo:Chess_qll45.svg** Fuente: https://upload.wikimedia.org/wikipedia/commons/9/9a/Chess_qll45.svg Licencia: CC-BY-SA-3.0 Colaboradores: Este gráfico vectorial fue creado con Inkscape. Artista original: en:User:Cburnett
- **Archivo:Chess_zhor_26.png** Fuente: https://upload.wikimedia.org/wikipedia/commons/4/4a/Chess_zhor_26.png Licencia: CC-BY-SA-3.0 Colaboradores: ? Artista original: ?
- **Archivo:Chess_zver_26.png** Fuente: https://upload.wikimedia.org/wikipedia/commons/d/dd/Chess_zver_26.png Licencia: Public domain Colaboradores: ? Artista original: ?
- **Archivo:Ocho_reinas_esquema.jpg** Fuente: https://upload.wikimedia.org/wikipedia/commons/9/90/Ocho_reinas_esquema.jpg Licencia: Public domain Colaboradores: ? Artista original: ?
- **Archivo:Ocho_reinas_reina_atacar_fila.JPG** Fuente: https://upload.wikimedia.org/wikipedia/commons/a/ad/Ocho_reinas_reina_atacar_fila.JPG Licencia: Public domain Colaboradores: ? Artista original: ?
- **Archivo:Ocho_reinas_reina_atacar.JPG** Fuente: https://upload.wikimedia.org/wikipedia/commons/d/d6/Ocho_reinas_reina_atacar.JPG Licencia: Public domain Colaboradores: ? Artista original: ?

8.3 Licencia de contenido

- Creative Commons Attribution-Share Alike 3.0