

REPÚBLICA BOLIVARIANA DE VENEZUELA

MINISTERIO DEL PODER POPULAR PARA LA EDUCACIÓN UNIVERSITARIA

INSTITUTO UNIVERSITARIO POLITÉCNICO DE LOS LLANOS “JUANA RAMIREZ
LA AVANZADORA”

PROGRAMA NACIONAL DE FORMACIÓN EN INFORMÁTICA

UNIDAD CURRICULAR: PROGRAMACIÓN Y ALGORITMÍA

SECCIÓN 2, TRAYECTO II

PROGRAMA DE LAS 8 REINAS USANDO LA TÉCNICA DE BACKTRACKING EN LA MÉTODOLOGÍA DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

FACILITADOR:

PROF. CARLOS PADRÓN

INTEGRANTES:

RENGIFO LIBARDO C.I: 2800010320

DICIEMBRE 2020

Introducción

El problema fue originalmente propuesto en 1848 por el ajedrecista Max Bezzel. Durante años, muchos matemáticos, incluyendo a Gauss y a Georg Cantor, han trabajado en él 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 $n \times n$ 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. Publicó una descripción muy detallada del desarrollo del algoritmo de backtracking, "depth-first".

El algoritmo planteado a continuación prone una solución con backtracking recursivamente usando una pila dinámica, la cual se ira llenando con los indices de las columnas a las cuales las reinas fueron colocadas, cada posición de la pila representa una fila dentro del tablero.

Con ello a través de unas meticulosos y sencillos cáculos mateáticos se puede comprobar la seguridad de una casilla dentro de la pila, así mismo se logra plantear la solución al problema utilizando la metodología de la prograamación orientada a objetos.

Puzzle de las 8 reinas

El **problema de las ocho reinas** es un pasatiempo que consiste en poner ocho reinas en el tablero de ajedrez sin que se amenacen. Fue propuesto por el ajedrecista alemán Max Bezzel en 1848.

Algoritmo

El algoritmo consiste en usar una pila dinámica de enteros, la cual se van a introducir los índices de las columnas de donde se encuentran cada reina dentro de un arreglo de enteros.

La pila se ira llenando de acuerdo a ciertos criterios, la reina no debe ser amenazada por las columnas, filas y diagonales.

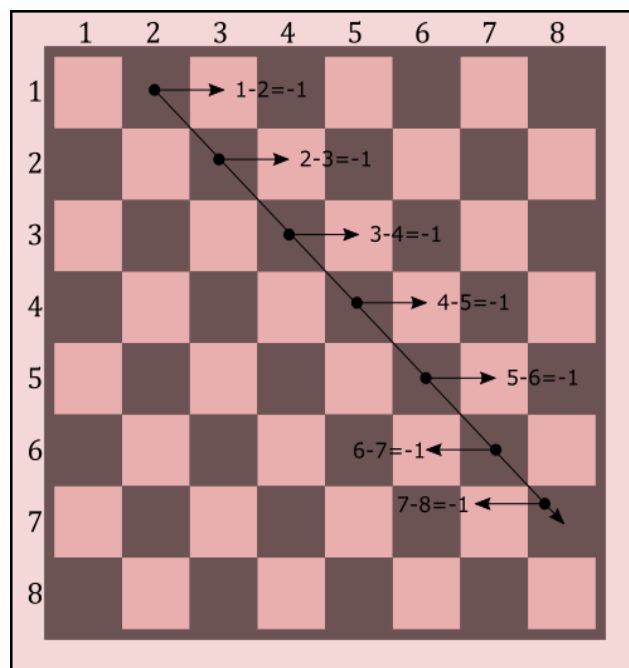
Para ello se realiza una comprobación sencilla para verificar las diagonales, se sigue la siguiente regla:

“Con esto en mente, se puede comprobar que en una misma diagonal descendente, todas las casillas cumplen que la diferencia entre el número de fila y el número de columna es constante: $i - j = k$. Siendo i el número de fila y j el número de columna.”

Para facilitar la comprensión del criterio

mencionado, se usa el valor absoluto de la resta de $i - j = k$, de esta forma se trabaja más fácil con dichas operaciones.

En el caso de las columnas, se comprueba por medio de un ciclo que itera hasta llegar al tamaño de la pila (con el objetivo de disminuir el rango de comprobación de columnas), si



la variable contadora (que representa una columna) es igual al índice de la columna almacenado en el arreglo entonces dicha columna esta ocupada por otra reina.

Si algunos de los 2 criterios mencionados anteriormente se cumplen, entonces no es posible colocar una reina en esa fila y columna.

El caso base para detenerla recursión del algoritmo, es aquella cuando la pila llegase a estar completamente vacía (**null**) y en dicha instancia se elimina las referencias de los objetos creados (pila y arreglo).

Si la pila llegase a estar llena es decir el tamaño de la pila es igual a la cantidad de reinas a buscar (**pila.size() == cantidadReinas**) , quiere decir que se encontro una solución sastifactoria.

Posteriormente de aplicar los pasos anteriores, se procede a eliminar la ultima dama en la pila y correrla una columna a la derecha, con el objetivo de seguir comprobando, y así que en algún punto, la pila llegue a estar nula (ya que la posicion de la primera dama no sastiface la colocación de las demás) y acabarse todas las soluciones posibles.

Para finalizar, se vuelve a invocar la función con los parámetros ya modificados y buscar otra solución que sastiface los criterios mencionados.

Implementación en el lenguaje java (clases)

El programa costa de una serie de clases y métodos para lograr la solución del problema, a continuación se hace mención de ellos con una breve explicación de su uso y sus métodos:

Acciones.java

Interfaz de la la cual las clases: Programa y Algoritmo heredan. Esta interfaz sirve para dar soporte a la 2 clases mencionadas ya que comparten un método en común, el de iniciar su algoritmo.

Helpers.java

Es una clase ayudante para facilitar el trabajo de algunas operaciones, como las siguientes:

- **imprimir(Object o)** se usa para imprimir en consola un objeto.
- **imprimirEnLinea(Object o)** se usa para imprimir sin salto de línea.
- **esNumero(String str)** se usa para saber si una cadena es un número.
- **numeroDentroDelRango (int n)** se usa para saber si $n \geq 1$ y $n \leq 8$.
- **leerCadena()** se usa para leer una cadena desde el teclado.

Programa.java

Clase que se utiliza para pedir datos, en este caso la cantidad de reinas que se van a usar dentro del tablero. Esta clase además, implementa un método heredado definido en la interfaz Acciones, dicho método tiene la lógica para pedir la cantidad de reinas y además esta comprobado para que dicha cantidad sea un valor numerico válido y comprendido entre 1-8.

Algoritmo.java

Es una clase la cual busca las soluciones de las 8 reinas por medio de una serie de métodos, los cuales son los siguientes:

- **comenzar()** iniciar la búsqueda de soluciones.
- **resolver(int columnaActual, int solucionesEncontradas, piladinámica <Integer> pila)** método recursivo que busca las soluciones, necesita la columna en la que va iniciar, sus soluciones encontradas hasta ahora y la pila.

- **esCorrectoElTablero(pila dinámica <Integer> pila, int columnaActual, int tablero[])** método que indica si una columna dada es adecuada para poner una reina, recibe la pila y el tablero, y por medio de un calculo matematico sencillo, se busca si dicha posicion es correcta.
- **imprimirSolucion(int solucion)** imprime el tablero con la solucion dada.
-

Reinas.java

Clase principal la cual el programa se ejecuta. Dicha clase ejecuta el método main el cual instancia un nuevo objeto aplicando el polimorfismo con la intefaz acciones, dicha instancia posee un método para iniciar el programa.

Conclusión

El algoritmo de las 8 reinas es un caso excepcional para aplicar técnicas de búsquedas como backtracking o fuerza bruta. Sin embargo ambas diferentes en términos de optimización y rapidez en cuanto a buscar las soluciones.

El algoritmo puede crearse recursivamente, de una forma más elegante y entendible o a través de ciclos, como en pocas ocasiones se suele emplear.

Una técnica muy importante para la resolución de problemas y implementaciones para la creación de inteligencias artificiales las cuales podrían evolucionar la vida cotidiana de las personas.