Análisis y diseño de algoritmos

Backtracking

Abel García Nájera

Universidad Autónoma Metropolitana Unidad Cuajimalpa

14 de julio de 2025

Objetivo

Objetivos de la clase

- Conocer el método de backtracking para resolver problemas.
- Aplicar backtracking para resolver algunos problemas.

Backtracking

Backtracking

A diferencia de la búsqueda exhaustiva, backtracking construye soluciones candidatas componente a componente y evalúa las soluciones parcialmente construidas:

> Si ningún valor potencial de los componentes restantes puede llevar a una solución, los componentes restantes no se generan en absoluto.

Este enfoque hace posible resolver algunos casos grandes de problemas combinatorios difíciles, aunque, en el peor caso, todavía nos enfrentamos al mismo tiempo exponencial encontrado en una búsqueda exhaustiva.

Árbol de espacio de estados

Se basa en la construcción de un árbol cuyos nodos reflejan elecciones específicas hechas para los componentes de una solución.

Los nodos del primer nivel en el árbol representan las elecciones hechas para el primer componente de una solución, los nodos del segundo nivel representan las elecciones para el segundo componente, y así sucesivamente.

Se dice que un nodo en un árbol de espacio de estados es prometedor si corresponde a una solución construida parcialmente que aún puede conducir a una solución completa, de lo contrario, lo llamamos no prometedor.

Las hojas representan callejones sin salida no prometedores o soluciones completas encontradas por el algoritmo.

Árbol de espacio de estados

En la mayoría de los casos, un árbol de espacio de estados para un algoritmo de backtracking se construye en la forma de búsqueda en profundidad:

Si el nodo actual es prometedor, su hijo se genera agregando la primera opción restante permitida para el siguiente componente de una solución y el procesamiento se traslada a este hijo.

Si el nodo actual resulta ser no prometedor, el algoritmo retrocede al padre del nodo para considerar la siguiente opción posible para su último componente. Si no existe tal opción, retrocede un nivel más en el árbol y así sucesivamente.

Finalmente, si el algoritmo alcanza una solución completa al problema, se detiene (si solo se requiere una solución) o continúa buscando otras posibles soluciones.

Problema de las N reinas

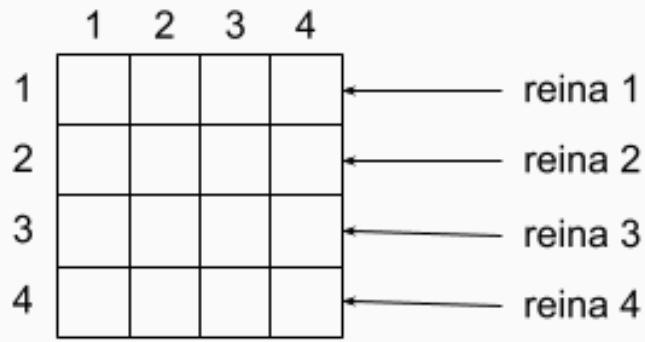
Problema de la N reinas

El problema consiste en colocar N reinas en un tablero de ajedrez de $N \times N$ para que no hayan dos reinas que se ataquen entre sí por estar en la misma fila o en la misma columna o en la misma diagonal.

Problema de la 4 reinas

Para $N = 1$, el problema tiene una solución trivial y es fácil ver que no hay solución para $N = 2$ y $N = 3$. Así que consideremos el problema de las cuatro reinas y lo resolveremos mediante la técnica de backtracking.

Dado que cada una de las cuatro reinas debe colocarse en su propia fila, lo que tenemos que hacer es asignar una columna para cada reina en el tablero.



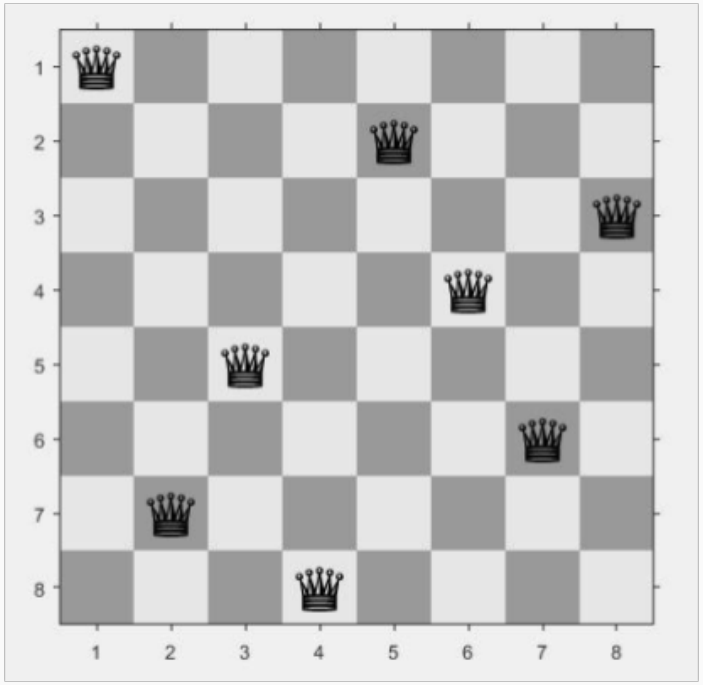
Para muchos problemas, el proceso de solución consiste en construirla mediante una secuencia de decisiones en la que cada opción lleva a futuras decisiones.

Si tomamos las decisiones correctas, obtendremos una solución.

Por otro lado, si después de haber tomado algunas decisiones llegamos a un callejón sin salida o nos damos cuenta de que no hemos hecho las decisiones adecuadas, tenemos que regresar a un punto de decisión anterior e intentar un camino diferente.

Problema de la N reinas

El problema consiste en colocar N reinas en un tablero de ajedrez de $N \times N$ para que no hayan dos reinas que se ataquen entre sí por estar en la misma fila o en la misma columna o en la misma diagonal.



Solución

Algoritmo N_REINAS (n, i)

```
1  si n = 0 entonces
2  |  regresar verdadero
3  si no
4  |  para j ← 1 hasta n hacer
5  |  |  si se puede colocar una reina en (i, j) sin peligro entonces
6  |  |  |  Colocar una reina en (i, j)
7  |  |  |  si N_REINAS (n − 1, i + 1) entonces
8  |  |  |  |  regresar verdadero
9  |  |  |  si no
10 |  |  |  |  Quitar reina de (i, j)
11 |  regresar falso
```

Sudoku

Problema de la N reinas

El objetivo del Sudoku es asignar dígitos a las celdas vacías para que cada fila, columna y subcuadrícula contenga exactamente una instancia de los dígitos del 1 al 9.

Las celdas iniciales se asignan para restringir el juego, de modo que solo haya una forma de terminarlo.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Solución

Algoritmo Sudoku (t)

```
1  si t no tiene celdas vacías entonces
2  |  regresar verdadero
3  si no
4  |  Obtener una celda vacía (i, j) de t
5  |  para d ← 1 hasta 9 hacer
6  |  |  si no hay conflicto para colocar d en (i, j) entonces
7  |  |  |  (i, j) ← d
8  |  |  |  si Sudoku (t) entonces
9  |  |  |  |  regresar verdadero
10 |  |  |  si no
11 |  |  |  |  (i, j) ← 0
12 |  regresar falso
```