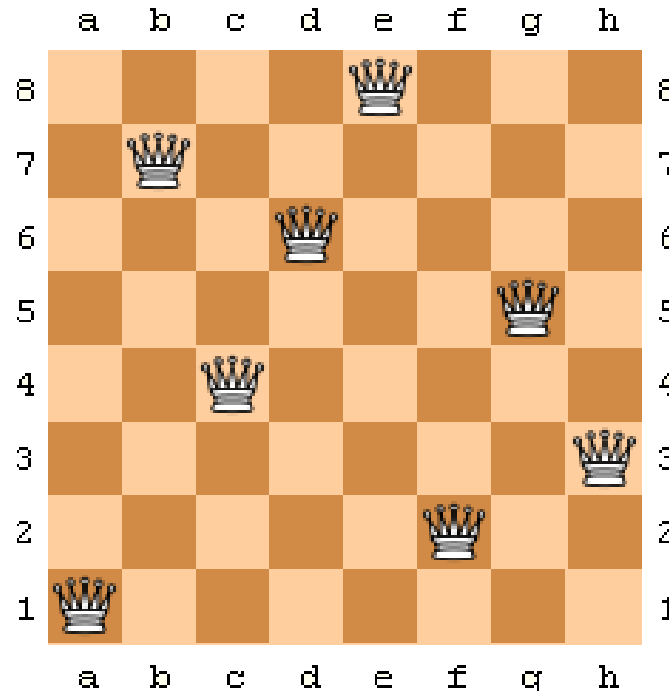


Problema de las ocho reinas y reducción del espacio de búsqueda

Problema de las ocho reinas

Consiste en acomodar 8 reinas en un tablero de ajedrez de tal manera que ninguna esté amenazada por ninguna otra. En ajedrez la reina amenaza a aquellas piezas que se encuentren en su misma fila, columna o diagonal y que no haya entre ellas otra pieza interfiriendo.

Una posible solución es:



Problema de las ocho reinas

¿Cuántas soluciones (acomodaciones de las 8 reinas) tiene este problema?

$64 \cdot 63 \cdot \dots \cdot 57 = 64! / 56! \approx 178 \text{ billones}$



¿Qué significa esto? Pues que en el mejor de los casos estaríamos hablando de tiempos de ejecución de minutos

¿Qué podemos hacer al respecto? Que tal reducir el espacio de búsqueda y, en ese espacio reducido, utilizar búsqueda exhaustiva.



Reducción del espacio de búsqueda

¿Por qué considerar que cada reina se puede mover en las 64 casillas del tablero y no, sabiendo que va a quedar una reina por fila, considerar las 8 posibilidades para cada una de ellas?

Haciendo esto, cada solución candidata se convierte en:

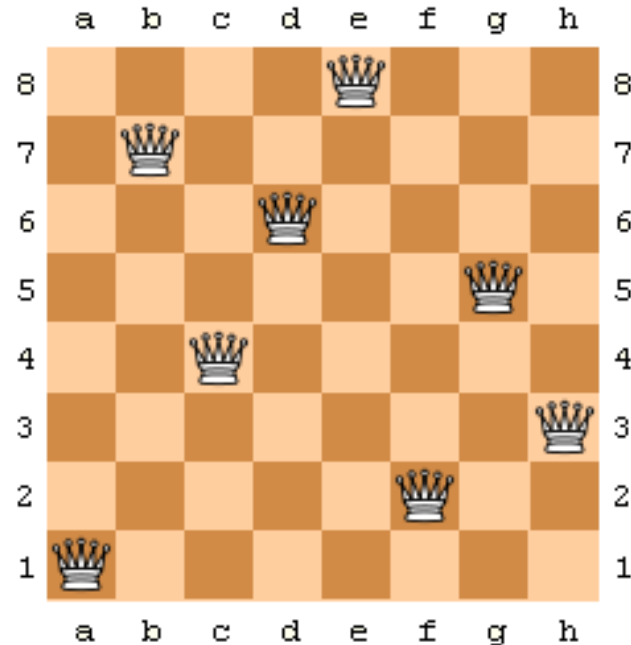
col R1	col R2	col R3	col R4	col R5	col R6	col R7	col R8
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Reducción del espacio de búsqueda

Como además dos reinas no pueden estar en la misma columna, en vez de 8^8 , solo hay en realidad $8!$ posibilidades que valga la pena analizar.

En otras palabras, el problema de las 8 reinas se convierte en un problema de las permutaciones de los números 1 a 8 que cumplan con la restricción

Por ejemplo, la solución anteriormente vista se puede representar como:



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

Solución mediante búsqueda exhaustiva

```
cont = 0
```

```
for i = 1 to 8:
```

```
    ci = i
```

```
do
```

```
    if true:
```

```
        process(c)
```

```
while nextPermutation(c)
```

```
print cont
```

```
process(c): //solo debe verificar diagonales
```

```
    for i = 1 to 7:
```

```
        p = 1
```

```
        for j = i+1 to 8:
```

```
            if cj = ci + p or cj = ci - p:
```

```
                exit
```

```
            p++
```

```
cont++
```

¿Cuál es la eficiencia de este algoritmo?

$(N+2)! = 2.580.480$ es decir, del orden de milisegundos

Reflexión

¿Cuándo debemos usar búsqueda exhaustiva?

Cuando sea estrictamente necesario evaluar todas los posibles candidatos en un problema

y/o

Cunado la cantidad de dichos candidatos sea pequeña