

# El Problema de las Ocho Reinas



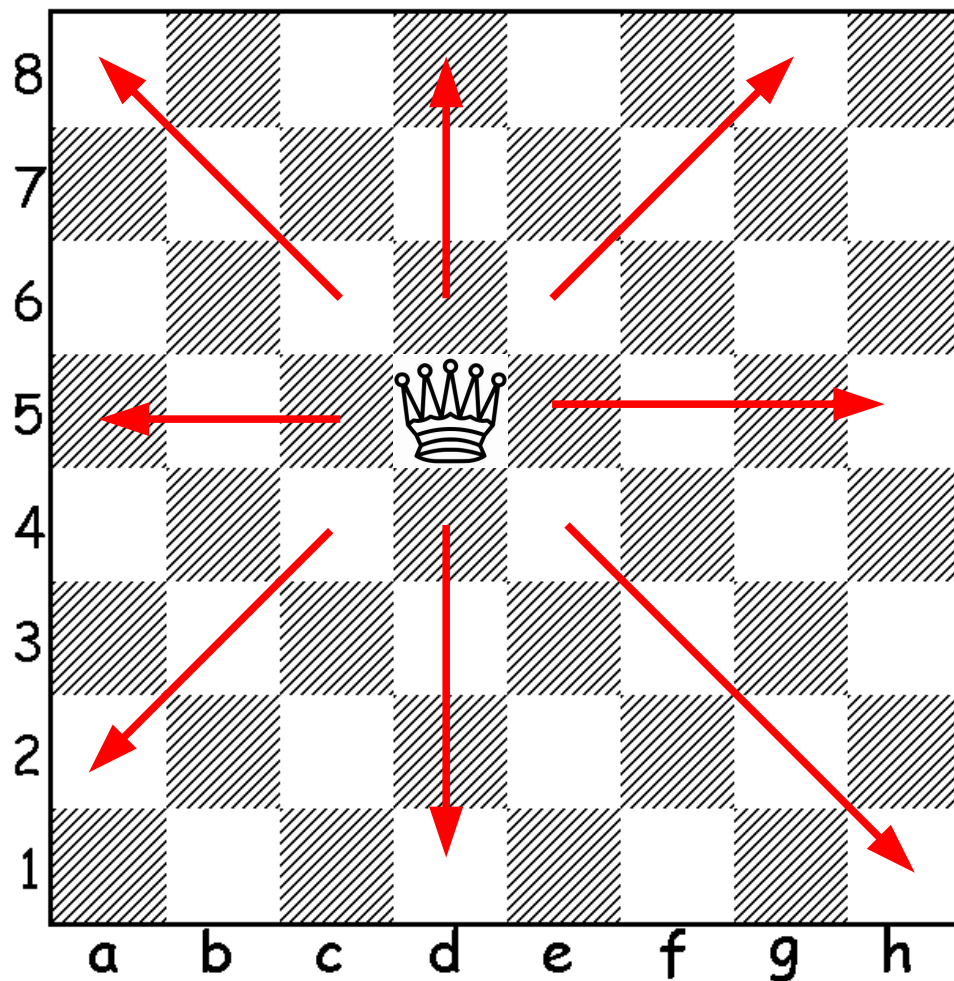
Rafael García Leiva

Email: [rafael.garcia@entropycs.com](mailto:rafael.garcia@entropycs.com)

Twitter: @rgarcialeiva

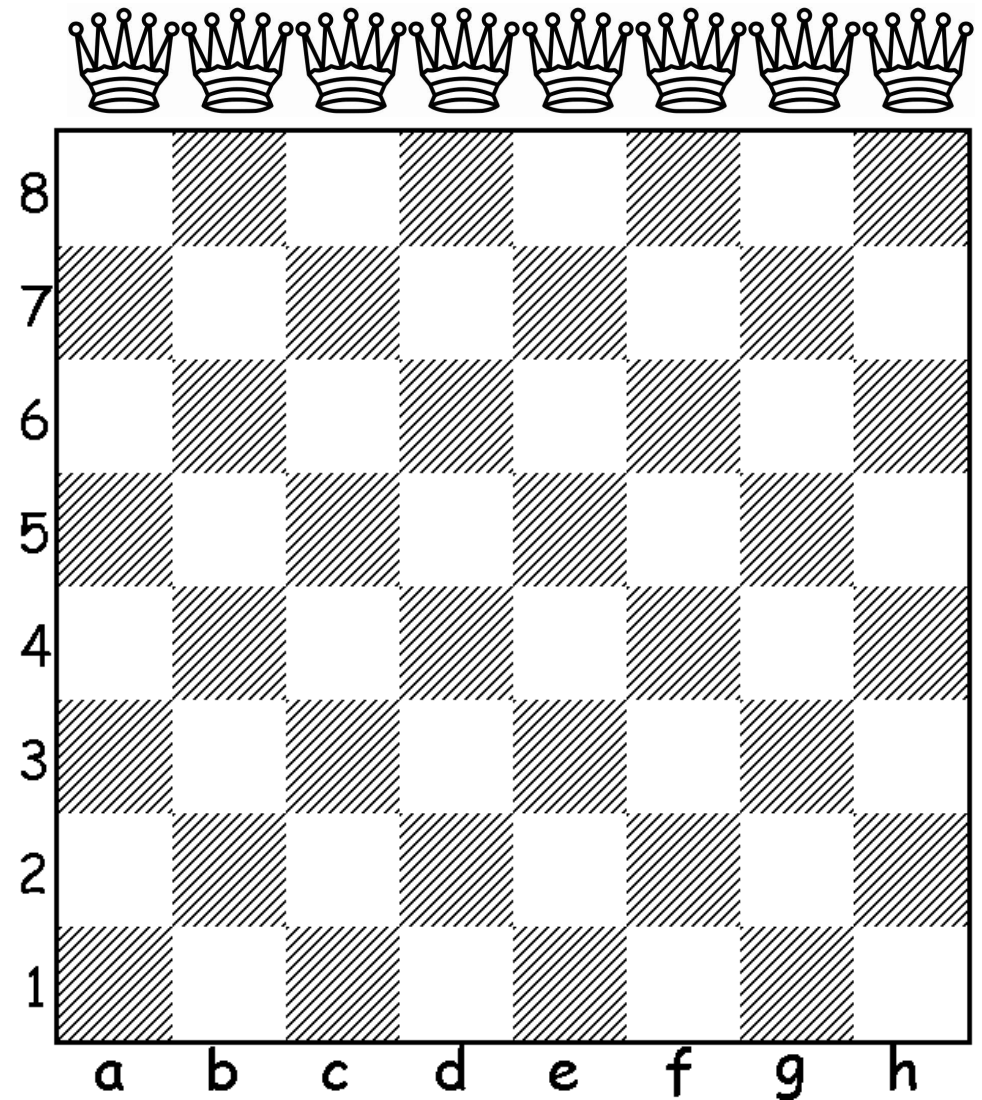
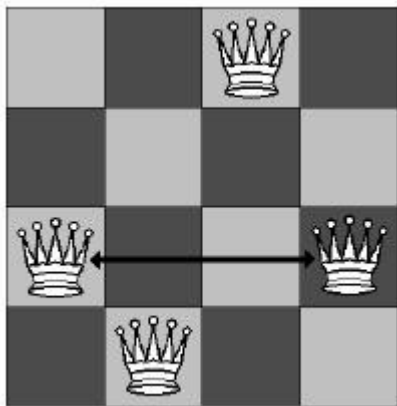
Blog: <http://mundoalgoritmos.blogspot.com.es/>

# Movimiento de la Reina en Ajedrez



# Descripción del problema

**Colocar ocho reinas en el tablero de tal manera que no se ataquen entre ellas**



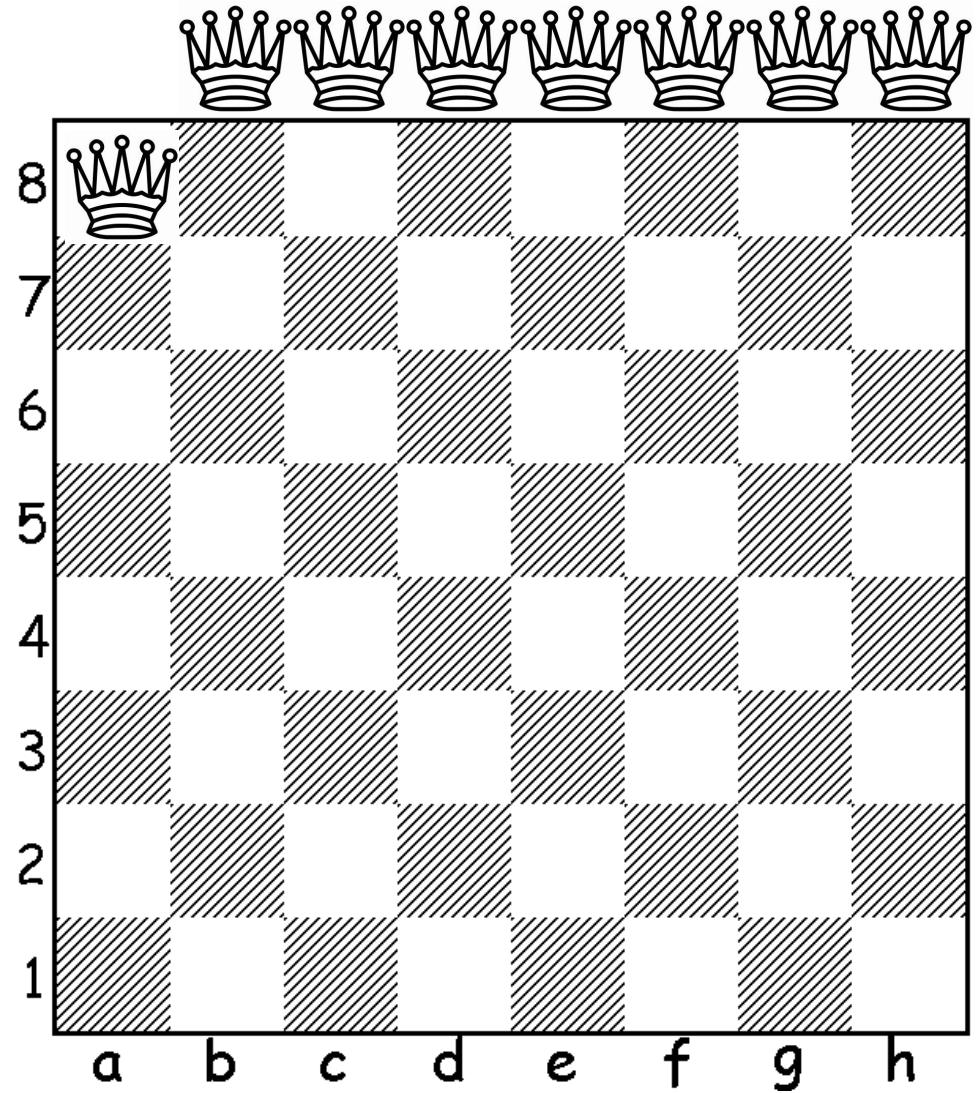
# Fuerza Bruta

## Fuerza bruta simple

$$64 \times 64 \times 64 \times 64 \times 64 \times 64 \times 64 \times 64 \\ = 281.474.976.710.656$$

**Mejora 1:** No colocar dos  
piezas en la  
misma casilla

$$64 \times 63 \times 62 \times 61 \times 60 \times 59 \times 58 \times 57 \\ = 178.462.987.637.760$$



# Fuerza Bruta

**Mejora 3:** Colocar una sola reina en cada columna

$$8 \times 8 \times 8 \times 8 \times 8 \times 8 \times 8 \times 8 \\ = 16.777.216$$

[1,1,1,1,1,1,1,1]

[1,1,1,1,1,1,1,2]

...

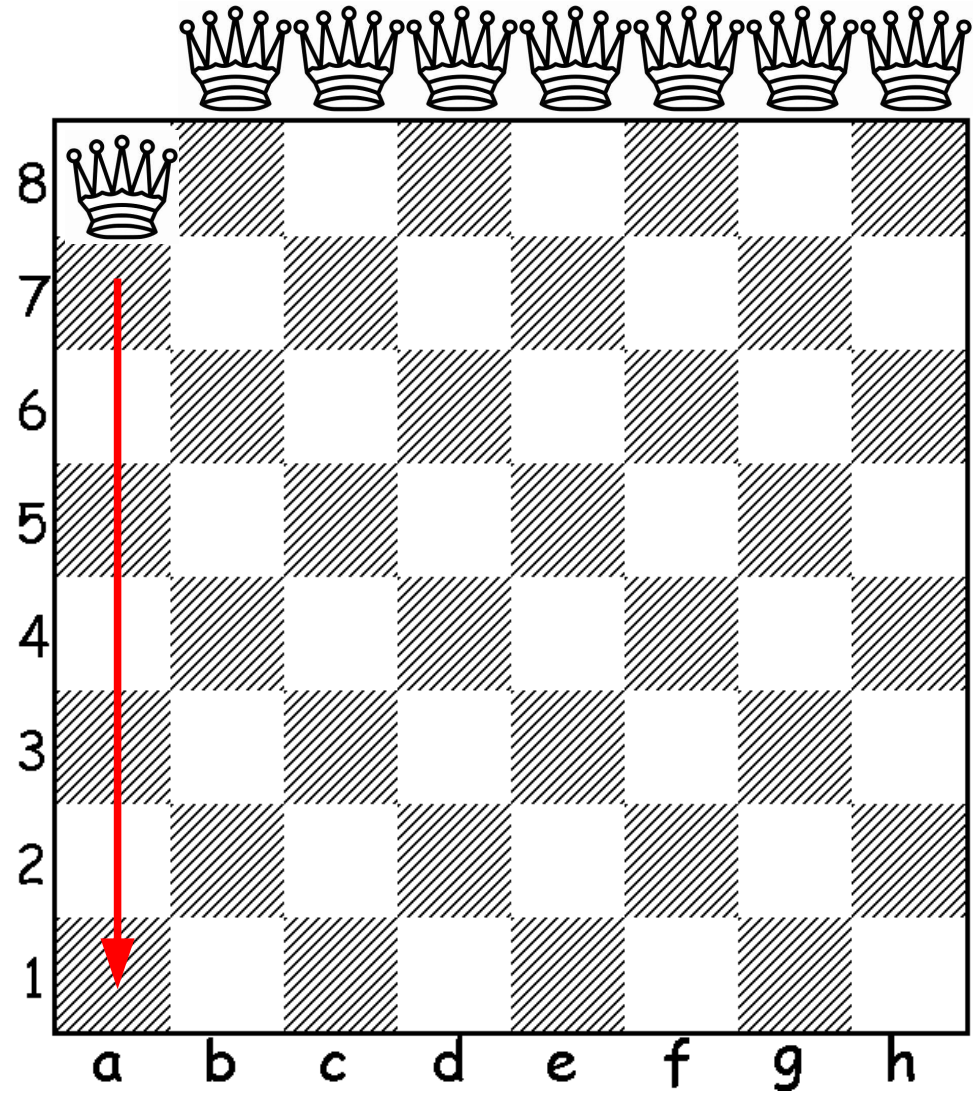
[1,1,1,1,1,1,2,1]

[1,1,1,1,1,1,2,2]

...

[8,8,8,8,8,8,8,7]

[8,8,8,8,8,8,8,8]



# Fuerza Bruta

**Mejora 4:** Colocar una sola reina en cada fila

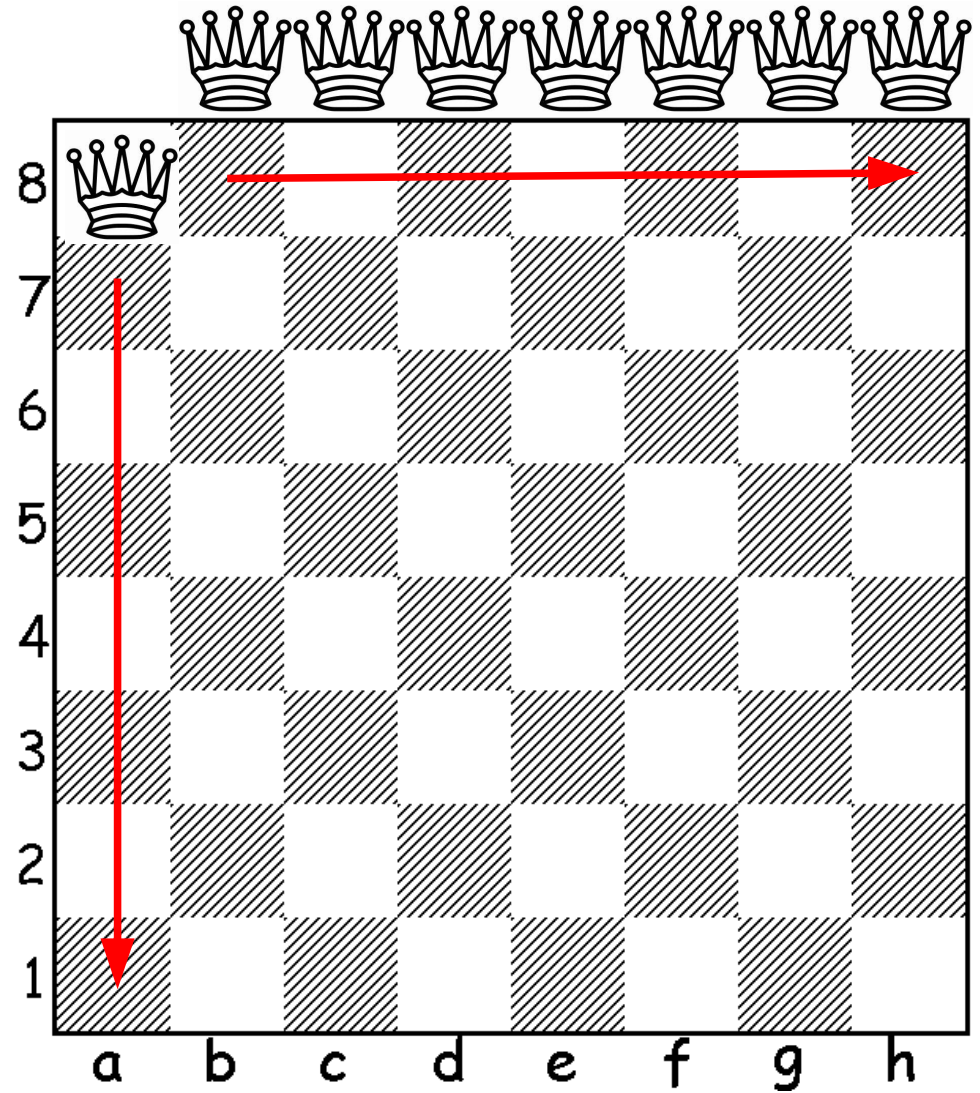
$$8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 \\ = 40.320$$

[1,2,3,4,5,6,7,8]

[1,2,3,4,5,6,8,7]

[1,2,3,4,5,8,6,7]

...

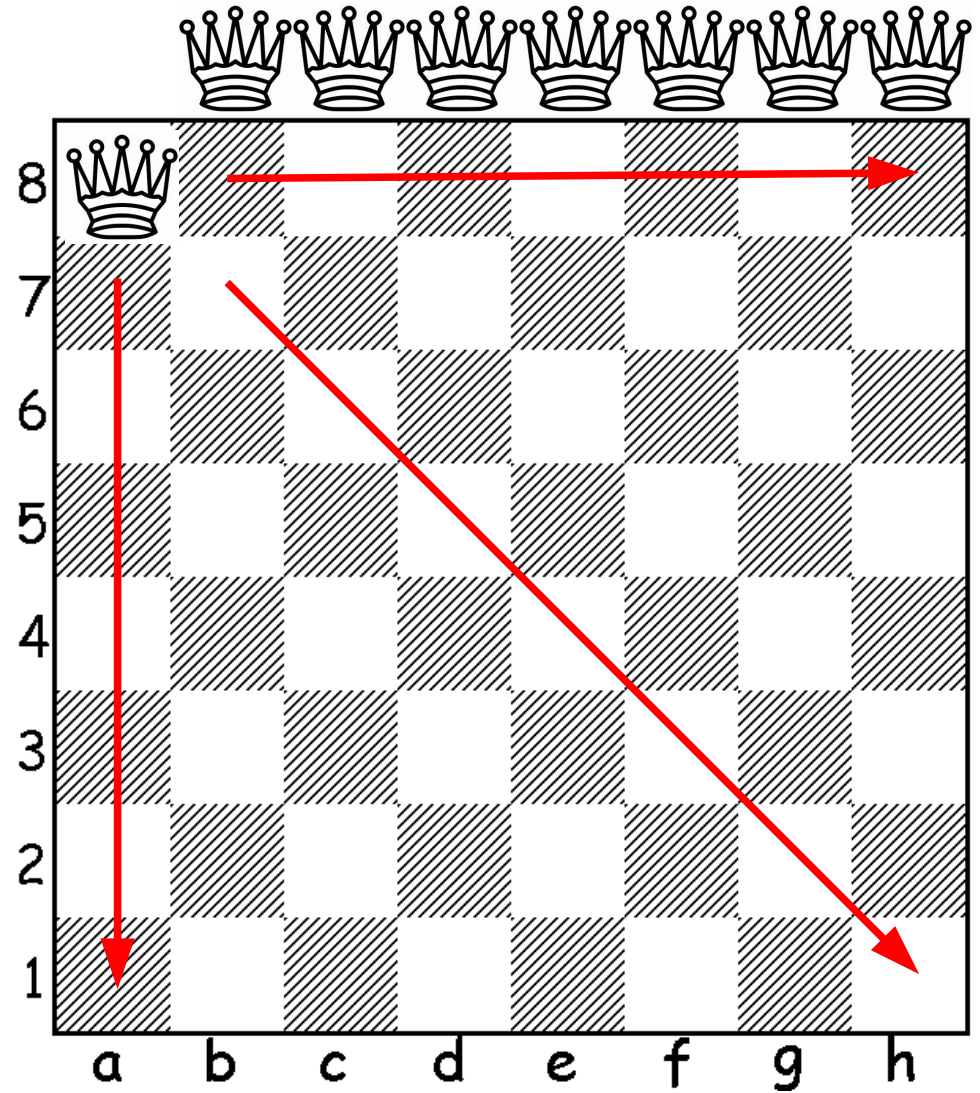


# Fuerza Bruta

**Mejora 5:** No colocar dos reinas en la misma diagonal

¿posibilidades?

¿algoritmo?



# Permutaciones en R

```
> library(combinat)
```

```
> permn(letters[1:3])
```

```
[[1]]
```

```
[1] "a" "b" "c"
```

```
[[2]]
```

```
[1] "a" "c" "b"
```

```
...
```

```
[[6]]
```

```
[1] "b" "a" "c"
```

```
> library(gtools)
```

```
> permutations(3, 3,  
letters[1:3])
```

```
      [,1] [,2] [,3]
```

```
[1,] "a"  "b"  "c"
```

```
[2,] "a"  "c"  "b"
```

```
[3,] "b"  "a"  "c"
```

```
[4,] "b"  "c"  "a"
```

```
[5,] "c"  "a"  "b"
```

```
[6,] "c"  "b"  "a"
```



# Algoritmo de Fuerza Bruta

```
library(combinat)
```

```
correcto <- function(p) {
```

```
  n <- length(p)
```

```
  for(i in 1:(n-1)) {
```

```
    for(j in (i+1):n) {
```

```
      if(abs(p[j] - p[i]) == abs(j - i))
```

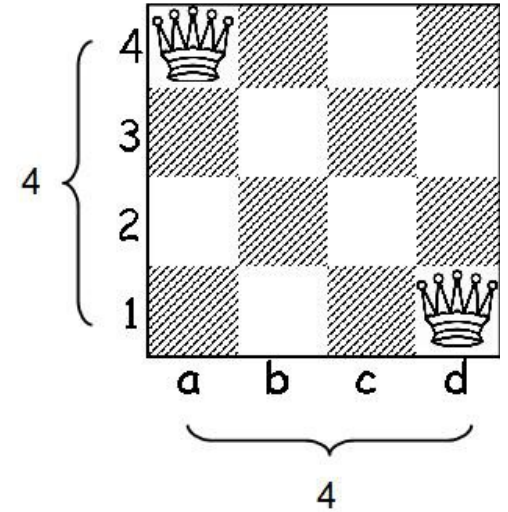
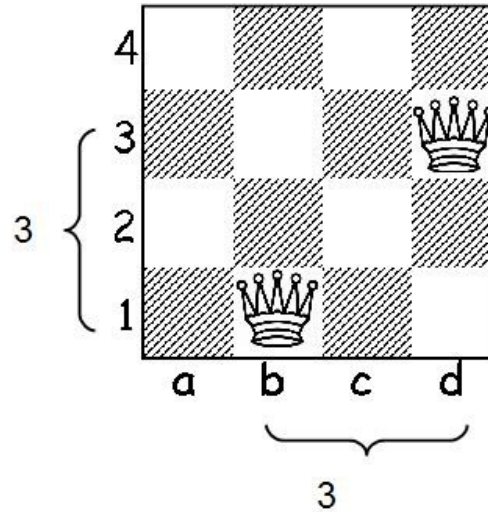
```
        return(FALSE)
```

```
    }
```

```
  }
```

```
  return(TRUE)
```

```
}
```

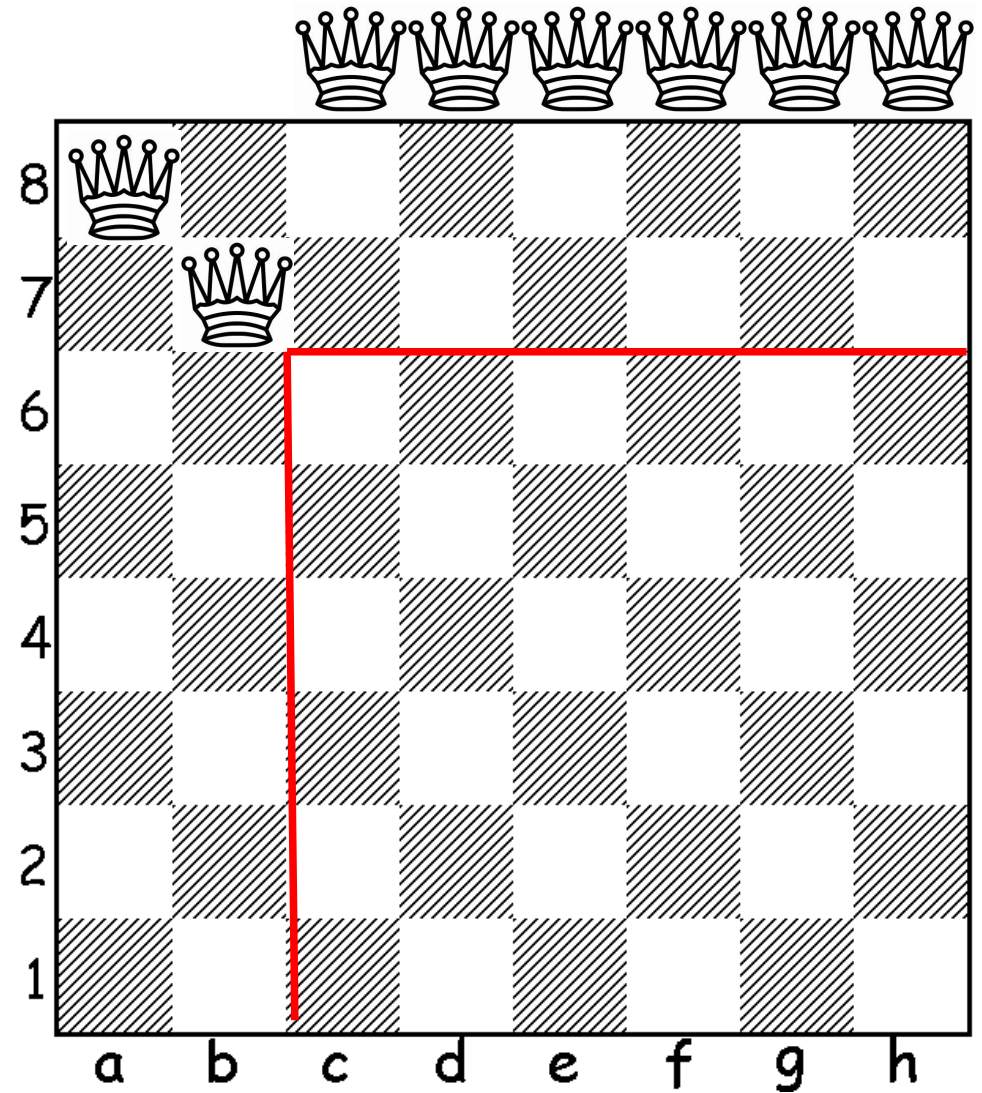


# Algoritmo de Fuerza Bruta

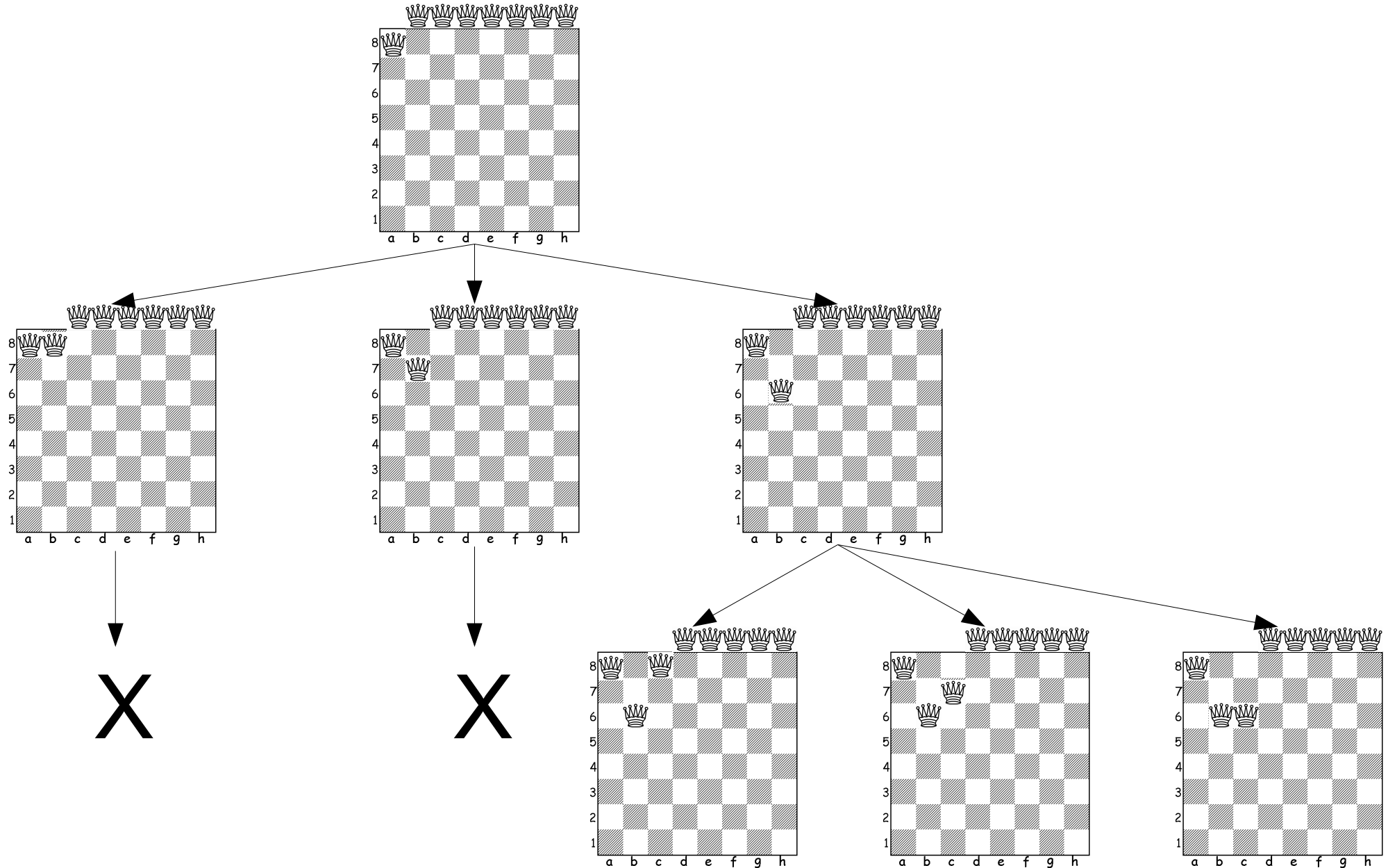
```
reinas <- function(n) {  
  permutaciones <- permn(1:8)  
  for(i in 1:length(permutaciones)) {  
    p <- permutaciones[[i]]  
    if(correcto(p)) {  
      cat(p, "\n")  
      k <- k + 1  
    }  
  }  
}
```

# La Técnica de la Vuelta Atrás

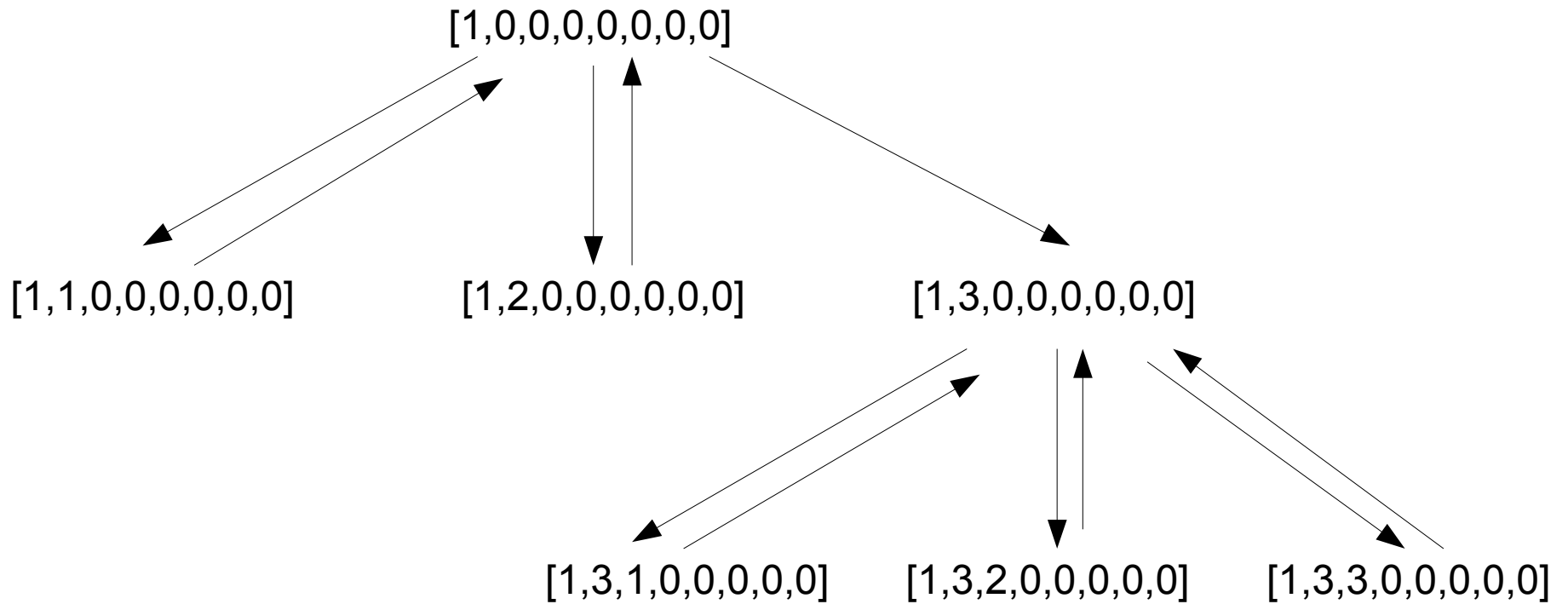
**No siempre es necesario  
construir completamente  
una solución para saber  
que esta es incorrecta**



# La Técnica de la Vuelta Atrás



# La Técnica de la Vuelta Atrás



# Algoritmo Vuelta Atrás

```
correcto <- function(x, k) {  
  if (k==1) return (TRUE)  
  
  for(i in 1:(k-1)) {  
    if( x[i] == x[k] ||  
        abs(x[i] - x[k]) == abs(i - k))  
      return(FALSE)  
  }  
  
  return(TRUE)  
}
```

# Algoritmo Vuelta Atrás

```
reinas <- function() {  
  
  x <- c(0, 0, 0, 0, 0, 0, 0, 0)  
  k <- 1  
  
  while (k >= 1) {  
    x[k] <- x[k] + 1  
    while (!correcto(x, k)) {  
      x[k] <- x[k] + 1  
      if (x[k]>8) break  
    }  
    if (x[k] > 8) {  
      k <- k - 1  
    } else {  
      if (k == 8) {  
        cat(x, "\n")  
      } else {  
        k <- k + 1  
        x[k] <- 0  
      }  
    }  
  }  
}
```

# Temas para Futuras Charlas

- Las torres de Hanoi
- El problema de la mochila
- El problema del viajante
- Ordenación súper rápida
- Los filósofos comensales
- Otros: sudoku, el famoso, la bandera holandesa, el juego del 15, asignación cuadrática, etc.