Cálculo de las N-reinas

Algoritmos Probabilísticos: Las Vegas

Daniel González Alonso

Índice

Algoritmos de las Vegas

- Algoritmos que usan decisiones probabilísticas para llegar más rápidamente a la solución correcta.
- Nunca retornan una solución incorrecta, pero pueden no llegar a dar una solución

Algoritmos de las Vegas

- Los algoritmos de las vegas pueden fallar, pero podemos repetir el proceso hasta dar con la solución.
- El numero de repeticiones que tendremos que realizar es 1/p(x), siendo p(x) la probabilidad de éxito.
- El tiempo en ejecutarse sigue la ecuación:

$$t(x) = p(x) * s(x) + (1 - p(x)) * (f(x) + t(x))$$

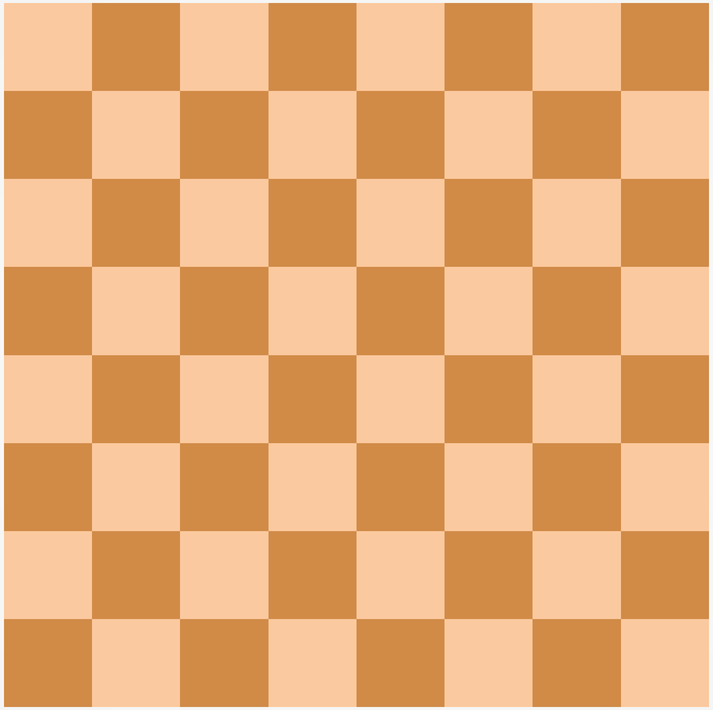
El puzle de las 8-reinas

- El puzle de las 8 reinas es un problema que consiste en colocar 8 reinas sin que se amenacen entre ellas en un tablero de ajedrez (8 filas y 8 columnas).
- Fue propuesto por primera vez en 1848 por Max Bezzel (ajedrecista), y las primeras soluciones publicadas por Franz Nauck dos años después.

El problema de las n-reinas

- El problema de las N-reinas es una generalización del problema anterior.
- Consiste en colocar un numero N (N > 3) de reinas en un tablero N*N sin que se amenacen entre ellas.

Solución mediante BackTracking



Encuentra la solución tras colocar 114 reinas.

Solución mediante Las Vegas

```
Function lasVegas(N):
    Entrada:
        N: dimensiones del tablero y numero de reinas.
    Salida:
        Lista con las reinas que resuelven el problema.

    reinas ← {}                                              // O(1)
    For  i←0 to N do                                         // O(1)
        posiciones ← posicionesPosibles(reinas, i)           // O(n*i)
        If (posiciones ≠ {}) then                            // O(1)
            selección ← U(0, longitud(posiciones))           // O(1)
            reinas ←reinas ∪ posiciones[selección]           // O(1)
        Else
            Return posiciones                                // O(1)
    Return reinas                                            // O(1)
```

¿Análisis de complejidad?

En caso de éxito:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{i} 1 = \sum_{i=1}^{n}\sum_{j=1}^{n} i = \sum_{i=1}^{n} n * i = n * \frac{n * (n+1)}{2}$$

Complejidad de orden: $O(n^2 * \frac{n+1}{2})$ = $O(n^3)$

Suponiendo el cálculo de números pseudoaleatorios en O(1).

Solución mediante Las Vegas

El numero esperado de nodos evaluados para las 8-reinas $(t(x))$ es:

$$\mathrm{t(x)} = s(x) + f(x) * \frac{1 - p(x)}{p(x)}$$

s(x) = 9
f(x) = 6,971 en promedio
p(x) = 0,1293

Al final: t(x) = 55,9424

Menos de la mitad que backtracking!

Solución alternativa

- Combinar backtracking con las Vegas (híbrido): se colocan las primeras piezas mediante las vegas y el resto por backtracking.