

ESTRUCTURA DE DATOS

ACTIVIDAD 3

Miguel Armando Rascon Dominguez



Universidad
Tecmilenio®

DISEÑO, IMPLEMENTACION Y FUNCIONAMIENTO

Introducción

El presente documento tiene como objetivo describir el diseño, la implementación y el funcionamiento de tres algoritmos fundamentales desarrollados en el lenguaje de programación Java:

1. Serie de Fibonacci recursiva
2. Suma de subconjuntos (Subset Sum)
3. Resolución de Sudoku mediante backtracking

Estos algoritmos fueron seleccionados por su relevancia en el aprendizaje de estructuras de control, recursión y técnicas de búsqueda sistemática. Java, como lenguaje orientado a objetos, ofrece un entorno robusto y ampliamente utilizado en entornos académicos y profesionales, lo que lo convierte en una excelente herramienta para comprender y aplicar conceptos algorítmicos. A lo largo del documento se presentará el diseño conceptual de cada problema, su implementación en código Java y una explicación detallada de su funcionamiento, con el fin de reforzar la comprensión de los principios de programación y su aplicación práctica.

Diseño

El problema consiste en calcular el n -ésimo número de la serie de Fibonacci, donde cada número es la suma de los dos anteriores, comenzando con 0 y 1. Se utiliza recursión para dividir el problema en subproblemas más pequeños hasta llegar a los casos base.

- **Entrada:** Un número entero n que indica la posición en la serie.
- **Salida:** El valor de Fibonacci en la posición n .
- **Casos base:**
 - $\text{fibonacci}(0) = 0$
 - $\text{fibonacci}(1) = 1$
- **Caso recursivo:**

- $\text{fibonacci}(n) = \text{fibonacci}(n-1) + \text{fibonacci}(n-2)$

Funcionamiento

1. El método fibonacci recibe un entero n.
2. Si n es 0 o 1, devuelve el valor directamente (caso base).
3. Si no, llama recursivamente a fibonacci(n-1) y fibonacci(n-2).
4. La recursión se resuelve sumando los resultados hasta llegar a los casos base.

2. Suma de Subconjuntos (Subset Sum)

Diseño

El objetivo es determinar si existe un subconjunto de un conjunto de enteros que sume un valor objetivo. Se utiliza recursión para explorar todas las combinaciones posibles de elementos.

- **Entrada:**
 - Un arreglo de enteros nums.
 - Un entero objetivo.
- **Salida:** true si existe un subconjunto cuya suma sea igual al objetivo, false en caso contrario.
- **Casos base:**
 - Si objetivo == 0, devolver true.
 - Si no hay más elementos o el objetivo es negativo, devolver false.
- **Caso recursivo:**
 - Incluir el elemento actual y verificar.
 - Excluir el elemento actual y verificar.

Funcionamiento

1. El método recibe el arreglo, el objetivo y el índice actual.
2. Si el objetivo llega a 0, se encontró un subconjunto válido.
3. Si se acaban los elementos o el objetivo es negativo, no hay solución en esa rama.
4. Se exploran dos caminos: incluir o excluir el elemento actual.
5. Si cualquiera de los caminos devuelve true, el subconjunto existe.

3. Resolución de Sudoku con Backtracking

Diseño

El Sudoku es un tablero 9x9 que debe completarse con números del 1 al 9 sin repetir en filas, columnas ni subcuadrículas 3x3. Se utiliza backtracking para probar números y retroceder si no cumplen las reglas.

- **Entrada:** Matriz 9x9 con ceros en las celdas vacías.
- **Salida:** Tablero resuelto.
- **Estrategia:**
 - **Buscar una celda vacía.**
 - **Probar números del 1 al 9.**
 - **Validar si el número es permitido.**
 - **Si es válido, colocarlo y continuar.**
 - **Si no se puede avanzar, retroceder (backtrack).**

Funcionamiento

- 1. El algoritmo busca la primera celda vacía.**
- 2. Prueba números del 1 al 9.**
- 3. Usa esValido para verificar que el número no esté repetido en la fila, columna o subcuadro.**
- 4. Si es válido, lo coloca y continúa recursivamente.**
- 5. Si no hay solución, retrocede y prueba otro número.**
- 6. Termina cuando no quedan celdas vacías.**