Actividad 1.2 Implementación de la técnica de programación "Programación dinámica" y "algoritmos avaros"

Realizado por alumnos de quinto semestre:

Luis Alejandro Carrillo Valdez - A01567228

Héctor Alfonso Vargas Carrera - A01563545

Instituto Tecnológico y de Estudios Superiores de Monterrey

Ingeniería en Tecnologías Computacionales

Campus Chihuahua

Análisis y diseño de algoritmos avanzados

(Gpo 600)

Mrta. Astrid E. Alcaraz

07 de septiembre del 2025

Chihuahua, Chihuahua

Actividad 1.2 Implementación de la técnica de programación "Programación dinámica" y "algoritmos avaros"

En la implementación siguiendo la programación dinámica, la complejidad en el peor caso es de O(V*N), donde V es la cantidad de vuelto que se tiene que dar y N siendo las diferentes denominaciones que el usuario va a elegir.

Esto se debe a que en el cálculo principal, que son 2 bucles anidados, se tiene un ciclo for donde se itera desde 1 hasta el valor del vuelto y genera V iteraciones. Luego tenemos que en el bucle interior se iteran todas las denominaciones que el usuario haya elegido, y esto genera N iteraciones. Este cálculo es el que causa el peor caso de complejidad, resultando en una complejidad de O(V*N).

En los otros casos que pueden causar complejidad es cuando se tiene que ordenar las denominaciones, lo cual tiene una complejidad de O(N log N). Además, se tiene otro donde se crea la variable monedas en un ciclo for con las denominaciones, lo cual es una complejidad de O(N).

El peor caso para este ejemplo viene del uso de 2 bucles anidados elevando la complejidad a una O(V*N).

En la implementación siguiendo el algoritmo avaro, la complejidad en el peor caso es de O(N), siendo N las diferentes denominaciones de monedas que el usuario elige.

Esto se debe a que en el cálculo inicial se requiere ordenar las denominaciones en orden descendente, lo cual genera un costo de O(N log N). Después, en el ciclo principal solo se recorre cada denominación una vez y se hacen operaciones constantes como división y módulo, lo que da una complejidad de O(N). También se tiene un costo de O(N) al inicializar la variable con las monedas y otro O(N) al imprimir los resultados.

En conclusión, el peor caso para este ejemplo se da por el proceso de buscar las denominaciones que se van a usar para el vuelto, resultando en una complejidad de O(N). Sin embargo, si se incluye también el ordenamiento, entonces sería de complejidad O(N log N) en el peor de los casos.

Casos de prueba empleados:

Caso 1:

Input: 3 denominaciones; 50, 100, 200; Precio a pagar 890; pagado 1000;

Output: No se tiene el vuelto suficiente

Caso 2:

Input: 5 denominaciones; 1, 2, 5, 10, 20; Precio a pagar 900; pagado 1000;

Output: 5 monedas de 20

Caso 3:

Input: 5 denominaciones; 1, 2, 5, 10, 20; Precio a pagar 800; pagado 800;

Output: No se necesita vuelto

Caso 4:

Input: 5 denominaciones; 1, 2, 5, 10, 20; Precio a pagar 800; pagado 600;

Output: Se ha pagado menos de lo requerido