

Explicación del Código: Vlad and a Sum of Digits

En este documento analizamos la implementación de dos enfoques para resolver el problema:

El método cándido (o ingenuo)

El método óptimo, basado en preprocesamiento

A ambos métodos se les entrega como parámetro un vector de casos, donde cada elemento representa un número para el cual Vlad desea conocer la suma acumulada de las sumas de dígitos desde 1 hasta ese valor.

1. Método Cándido

La solución ingenua recalculara desde cero la suma acumulada de dígitos para cada caso del vector. Es decir, si un caso pide el valor para $n = 50$, el método vuelve a sumar los dígitos de 1, 2, 3... hasta 50, aunque otros casos puedan haber pedido valores similares previamente.

Estructura de la solución

1. Crear un arreglo de resultados, donde se almacenará la respuesta final de cada caso.
2. Para cada número del vector de entrada:
 - Se construye una secuencia auxiliar, que almacenará las sumas acumuladas desde 1 hasta el valor del caso actual.
Esta clase Secuencia ya está definida en el módulo de estructuras lineales del proyecto.
3. Inicialización de la secuencia
Como la clase Secuencia rellena con null y no con cero, primero la llenamos completamente con 0 utilizando su método insertar.
4. Cálculo de la suma de dígitos
Para cada número k entre 1 y el valor del caso:
 - Se extraen sus dígitos usando un ciclo while que:
 - Suma $n \% 10$
 - Avanza con $n = n / 10$
 - Al terminar, se agrega la suma de dígitos al acumulado anterior de la secuencia.
 - El resultado parcial se almacena mediante set en la posición correspondiente.
5. Guardar la respuesta
Una vez terminada la secuencia, se toma el valor de la última posición —la que corresponde al número del caso— y se agrega al vector resultado.

Limitaciones

- Recalcula todo desde cero para cada caso.
- Si varios casos incluyen valores similares, se repite el mismo trabajo una y otra vez.
- Correcto pero poco eficiente.

2. Método Óptimo

En lugar de recalcular toda la secuencia para cada caso, el método óptimo crea una única secuencia auxiliar global que contiene las respuestas para todos los números desde 1 hasta el máximo posible permitido por el problema:

$\text{max} = 2 \times 10^5$

Este preprocesamiento se hace una sola vez, y luego cada caso se responde en tiempo constante.

Variables principales

- max: límite superior permitido por el problema (200 000).
- B: secuencia auxiliar de tamaño $\text{max} + 1$, donde se almacenarán todos los resultados acumulados.

3. Llenado de la Secuencia Auxiliar

Este procedimiento realiza los mismos cálculos del método cándido, pero solo una vez y para todos los valores desde 1 hasta max.

Pasos:

1. Inicializar la secuencia con ceros, igual que en el método cándido.
2. Para cada valor i entre 1 y max:
 - Calcular la suma de dígitos de i con el mismo procedimiento basado en módulo y división.
 - Sumarla al valor acumulado de la posición anterior.
 - Guardar el resultado en la posición i .

Cuando este proceso termina, la secuencia B contiene todas las respuestas posibles.

4. Cómo Responde el Método Óptimo

Para cada valor del vector de casos:

- Se consulta directamente la posición correspondiente en B.
- Se agrega ese valor al arreglo de salida.

De esta forma, cada caso se resuelve en $O(1)$.

5. Resultado y Comparación

Ambos métodos producen respuestas correctas.

La diferencia está en el rendimiento:

- El método cándido recalcula todo para cada caso \Rightarrow más lento.
- El método óptimo precalcula una sola vez y responde casi instantáneamente \Rightarrow mucho más eficiente.

Este patrón es muy común en problemas de programación competitiva:
preprocesar una vez para responder muchas consultas en tiempo constante.