Explicación del Problema: Vlad and a Sum of Digits

En este problema, proveniente de la plataforma Codeforces, se trabaja con la transformación de números mediante la suma de sus dígitos. El objetivo es calcular, para cada caso de prueba, la suma total obtenida después de transformar todos los números desde 1 hasta N aplicando este procedimiento.

1. Enunciado del Problema

Vlad escribe todos los números desde 1 hasta N. Luego reemplaza cada número por la suma de sus dígitos.

Por ejemplo, si N = 12, la secuencia original es:

1, 2, 3, ..., 12

Cada número se transforma en la suma de sus dígitos:

- $1 \rightarrow 1$
- $2 \rightarrow 2$
- …
- $9 \rightarrow 9$
- $10 \rightarrow 1 + 0 = 1$
- $11 \rightarrow 1 + 1 = 2$
- $12 \rightarrow 1 + 2 = 3$

Por lo tanto, la secuencia transformada queda así:

1 2 3 4 5 6 7 8 9 1 2 3

La suma total es:

1 + 2 + ... + 9 + 1 + 2 + 3 = 51

Este es exactamente el valor que debemos calcular.

2. Entradas y Salidas

Entrada

- Un entero T ($1 \leq T \leq 10^4$), que indica cuántos casos de prueba hay.
- Luego, para cada caso:
  - Un entero N ($1 \leq N \leq 2 \times 10^5$).

Salida

Para cada caso, imprimir la suma acumulada de los valores transformados desde 1 hasta N.

3. Problemas de eficiencia

Una solución básica es calcular la suma desde cero para cada uno de los casos.

Cómo funciona

Para un N dado:

1. Se recorre cada número desde 1 hasta N.
2. Se obtiene la suma de dígitos de ese número.
3. Se acumula el resultado en una variable.
4. Una vez terminado el recorrido, se imprime el resultado.
5. Este proceso se repite completamente para todos los casos.

Ejemplo

Si la entrada incluye dos casos:

- N = 12
- N = 12

El algoritmo vuelve a calcular todos los valores del 1 al 12 dos veces, repitiendo exactamente el mismo trabajo.

Desventaja

Esta forma es correcta, pero:

- Repite cálculos idénticos.
- Es ineficiente si T es grande.
- El tiempo de ejecución crece notablemente con valores grandes de N.

Su complejidad aproximada es $O(T \times N \times \log N)$ (por el costo de sumar dígitos), lo cual puede llegar a sentirse como $O(N^3)$ en los peores análisis empíricos del comportamiento.

4. Solución Óptima: Preprocesamiento

La clave para mejorar el rendimiento está en no repetir cálculos.

Idea principal

Antes de procesar cualquier caso:

1. Se construye un arreglo sumas[] de tamaño igual al máximo valor permitido de N (200 000).
2. Este arreglo almacena en cada posición i la suma total de los valores transformados desde 1 hasta i.
3. El preprocesamiento se realiza solo una vez.

Luego, para cada caso:

- Simplemente se imprime sumas[n].
- Esto toma tiempo constante O(1).

Ventajas

- Los cálculos complejos se hacen una sola vez.
- Cada consulta es rápida.
- Ideal cuando T es grande.

Complejidad

- El preprocesamiento toma O(N).
- Cada consulta toma O(1).
- Complejidad total: O(N + T).

Esta estrategia es ampliamente superior a recalcular todo cada vez.

5. Conclusión

El problema consiste en transformar cada número del 1 al N mediante la suma de sus dígitos y acumular el resultado. Aunque es posible hacerlo directamente para cada caso, esto resulta muy lento cuando hay muchos casos o valores muy grandes.

Por ello, la mejor estrategia es:

- Realizar un preprocesamiento único,
- Guardar los resultados parciales,
- Responder consultas en tiempo constante.

Esto vuelve la solución óptima altamente eficiente comparada con la ingenua.