

Explicación del Código: Hoteles a lo largo de la costa croata

En este documento explicamos paso a paso cómo se implementan dos soluciones en Java para resolver el problema de encontrar la mayor suma posible de precios de hoteles consecutivos sin superar un límite M . Las dos soluciones son: una versión sencilla (cándida o ingenua) y una versión óptima usando la técnica Sliding Window.

1. Método Cándido (Enfoque Básico)

Este método recorre todas las posibles combinaciones de hoteles consecutivos iniciando desde cada posición.

Para ello se utilizan dos variables:

- suma: almacena la suma de la combinación actual.
- maximo: almacena el mayor valor de suma encontrado sin exceder M .

El algoritmo:

- Para cada posición i en el arreglo de hoteles:
- Se asigna $\text{suma} = \text{hoteles}[i]$.
- Desde esa posición se recorre hacia adelante con un segundo ciclo.
- Se acumulan valores consecutivos mientras no se supere M .
- Cuando una suma válida es mayor que el maximo, se actualiza.

Finalmente, se retorna maximo.

Este método funciona, pero es lento porque utiliza ciclos anidados. Su complejidad es $O(n^2)$.

2. Método Óptimo (Sliding Window)

Este método mejora drásticamente la eficiencia utilizando dos punteros:

- izquierda (L)
- derecha (R)

Además utiliza suma y maximo igual que el método anterior.

El funcionamiento:

1. Mientras R sea menor que la cantidad de hoteles:

- Si sumar $\text{hoteles}[R]$ a la suma actual no supera M :
 - Se agrega ese valor a suma.
 - Se actualiza maximo si corresponde.
 - Se incrementa R para ampliar la ventana.
- Si la suma excede M al intentar agregar $\text{hoteles}[R]$:
 - Se resta el valor de $\text{hoteles}[L]$ de la suma.
 - Se incrementa L para reducir la ventana.

2. Al terminar, se retorna maximo.

Gracias a esta técnica, la lista se recorre solo una vez. Su complejidad es $O(n)$.

3. Comparación de Rendimiento

Utilizando la clase Stopwatch para medir tiempos, se observa una gran diferencia:

- Con 240,000 hoteles:
 - Método cándido: ~3000 microsegundos.
 - Método óptimo: ~83 microsegundos.

Esto demuestra la superioridad del enfoque Sliding Window.

Conclusión:

Ambos métodos encuentran la respuesta correcta, pero el método óptimo es muy superior al reducir el tiempo de ejecución de forma significativa. Esta mejora es fundamental en problemas donde N puede ser muy grande.