Coin Collector (UVA 11264)

Versión Alternativa del Problema

Nuestro querido Sultán está visitando un país donde existen n tipos diferentes de monedas.

Él desea coleccionar la mayor cantidad posible de tipos diferentes de monedas.

Si desea retirar una cantidad X de dinero del banco, el banco usará el siguiente algoritmo:

Algoritmo del banco

```
withdraw(X){
 if (X == 0) return:
 Sea Y la moneda de mayor valor que no exceda X
 Entregar al cliente una moneda de valor Y
withdraw(X - Y);
```

El Sultán puede retirar cualquier cantidad de dinero.

Debe maximizar la cantidad de tipos de monedas diferentes que puede obtener en un solo retiro.



Ejemplo

Entrada:

6 monedas: 1 2 4 8 16 32

4 monedas: 1 3 6 8

Salida:

6

> 3

Explicación:

- ► En el primer caso se pueden tomar todas.
- ► En el segundo caso se pueden tomar 1, 3 y 8.

Idea General de la Solución

Este problema se puede resolver con un enfoque **greedy** (codicioso):

- Siempre se elige una moneda si la suma de las monedas elegidas hasta el momento más la actual es menor que la siguiente.
- Siempre se puede tomar la última moneda.

Algoritmo Paso a Paso

Supongamos que tenemos monedas c_1, c_2, \ldots, c_n :

- 1. Inicializar suma = 0, contador = 0.
- 2. Para cada i desde 0 hasta n-2:
 - ▶ Si suma + c_i < c_{i+1} , incluir c_i .
- 3. Incluir siempre la última moneda c_n .

Complejidad del Algoritmo

- **Tiempo:** $\mathcal{O}(n)$ por cada caso de prueba.
- **Espacio:** $\mathcal{O}(1)$ si no contamos el arreglo de entrada.

Muy eficiente incluso para *n* grande.