

Optimizando monedas

Contribución de Agustín Santiago Gutiérrez

Descripción del problema

Bernardo el vendedor tiene que dar un vuelto. Esto requiere rápidos cálculos mentales para dar el vuelto con la mínima cantidad de monedas posible. Más importante aún, esto requiere gran destreza manual para manipular las monedas, y ese no es el fuerte de Bernardo.

Existe una lista de N denominaciones de monedas, de a_1 centavos, a_2 centavos, ..., a_{N-1} centavos y a_N centavos. Bernardo posee un suministro ilimitado de monedas de cada una de estas denominaciones.

Bernardo utilizó lo que aprendió estudiando robótica en la universidad pública gratuita y de calidad, para programar y construir un robot automático de entrega de vuelto que lo ayude en su negocio. Programó al robot utilizando un algoritmo denominado goloso, avaro o voraz (se les llama *greedy* en inglés). En el algoritmo, se consideran las denominaciones disponibles a_i , y el monto restante de V centavos que falta entregar. Se elige en cada paso la más grande posible a_k tal que $a_k \leq V$, es decir sin pasarse del vuelto restante por entregar V . Se agrega una moneda de esa denominación a_k a la solución, y se resta su valor al vuelto restante V , es decir se realiza la asignación $V \leftarrow V - a_k$. El algoritmo continúa hasta que sea $V = 0$.

Tu tarea consiste en ayudar a Bernardo a programar una función que le indique si su programa es correcto, y si no lo es, le indique el menor posible de V para el cual su programa no utiliza la mínima cantidad posible de monedas.

Descripción de la función

Se debe implementar una función `monedas(a)`, que reciba un arreglo a de N enteros con las denominaciones de monedas a_i .

La función debe retornar un único entero positivo que indique el mínimo valor de V tal que el algoritmo de Bernardo falla si se lo manda a entregar un vuelto V (es decir, no entrega la mínima cantidad de monedas posible para dar un vuelto V). Si para las denominaciones indicadas

el algoritmo de Bernardo es correcto, se debe retornar -1 .

Evaluador local

El evaluador local lee de la entrada estándar:

- Entero N
 - N enteros $a[i]$
- Escribe en la salida un único entero retornado por la función `monedas`.

Restricciones

- $1 \leq N \leq 5000$
- $a_1 = 1, a_i < a_{i+1} \leq 10^4$

Ejemplos

Si la entrada es:

```
4
1 5 11 20
```

La salida correcta es:

```
15
```

Ya que $V = 15$ es el primer fallo del algoritmo de Bernardo: usará cinco monedas formando $15 = 11 + 1 + 1 + 1 + 1$, pero se puede hacer $15 = 5 + 5 + 5$ usando solamente tres monedas.

Si en cambio es:

```
4
1 5 15 30
```

La salida es:

```
-1
```

Subtareas

1. $N = 3, a_2 = 3$ (11 puntos)
2. $N \leq 3$ (20 puntos)
3. $N = 8$, siete de los números de a son $1, 2, 5, 10, 25, 50, 100$ (22 puntos)
4. $N \leq 10, a_i \leq 50$ (32 puntos)
5. Sin restricciones adicionales (15 puntos)