

Empacotando biscoitos (biscuits)

Tia Khong está organizando uma competição com x participantes, e deseja dar a cada participante um **saco de biscoitos**. Há k diferentes tipos de biscoitos, numerados de 0 até $k - 1$. Cada biscoito de tipo i ($0 \leq i \leq k - 1$) tem um **valor de deliciosidade** de 2^i . Tia Khong tem $a[i]$ (possivelmente zero) biscoitos de tipo i na sua despensa.

Cada saco de Tia Khong conterá zero ou mais biscoitos de cada tipo. O número total de biscoitos de tipo i em todos os sacos não pode exceder $a[i]$. A soma dos valores de deliciosidade de todos os biscoitos em um saco é chamado de **deliciosidade total** do saco.

Ajude Tia Khong a determinar quantos valores de y existem, tal que é possível empacotar x sacos de biscoitos, cada saco com deliciosidade total igual a y .

Detalhes de implementação

Você deve implementar o seguinte procedimento:

```
int64 count_tastiness(int64 x, int64[] a)
```

- x : o número de sacos de biscoitos a empacotar.
- a : um array de comprimento k . Para $0 \leq i \leq k - 1$, $a[i]$ denota o número de biscoitos de tipo i na despensa.
- O procedimento deve retornar o número de diferentes valores de y , tal que a Tia pode empacotar x sacos de biscoitos, cada um tendo uma deliciosidade total de y .
- O procedimento é chamado um total de q vezes (veja as seções Restrições e Subtarefas para os valores permitidos de q). Cada uma dessas chamadas deve ser tratada como um cenário separado.

Exemplos

Exemplo 1

Considere a seguinte chamada:

```
count_tastiness(3, [5, 2, 1])
```

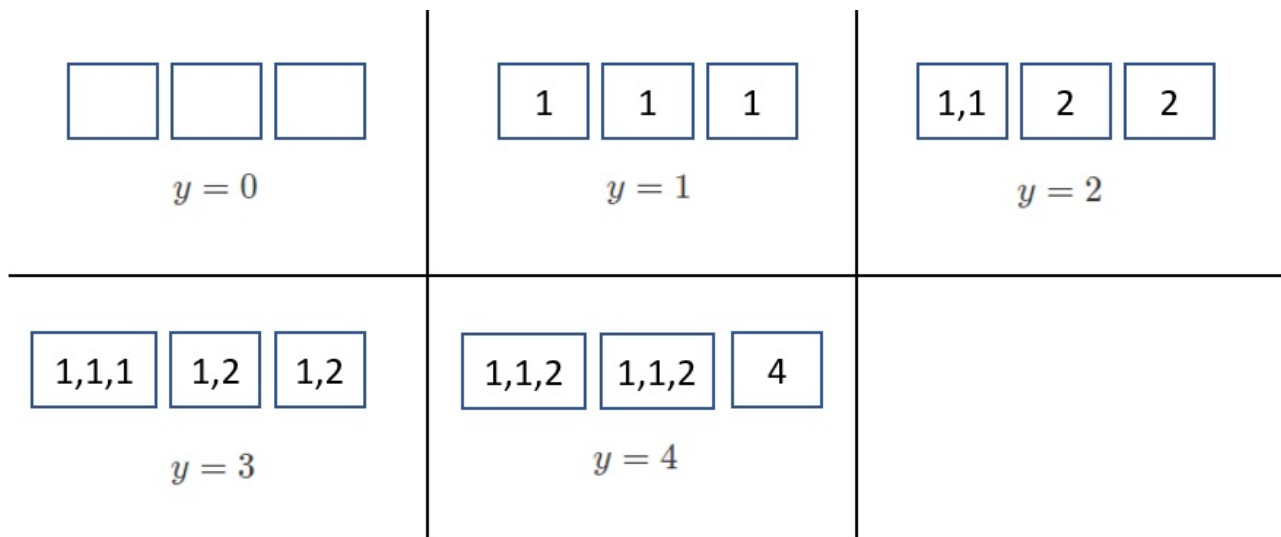
Isso significa que a Tia quer empacotar 3 sacos, e há 3 tipos de biscoitos na despensa:

- 5 biscoitos de tipo 0, cada um tendo valor de deliciasidade 1,
- 2 biscoitos de tipo 1, cada um tendo valor de deliciasidade 2,
- 1 biscoito de tipo 2, com valor de deliciasidade 4.

Os valores possíveis de y são $[0, 1, 2, 3, 4]$. Por exemplo, para empacotar 3 sacos de deliciasidade total 3, a Tia pode empacotar:

- um saco contendo três biscoitos de tipo 0, e
- dois sacos, cada um contendo um biscoito de tipo 0 e um biscoito de tipo 1.

Como há 5 valores possíveis de y , o procedimento deve retornar 5.



Considere a seguinte chamada:

```
count_tastiness(2, [2, 1, 2])
```

Isso significa que a Tia deseja empacotar 2 sacos, e há 3 tipos de biscoitos na despensa:

- 2 biscoitos de tipo 0, cada um tendo valor de deliciasidade 1,
- 1 biscoito de tipo 1, com valor de deliciasidade 2,
- 2 biscoitos de tipo 2, cada um tendo valor de deliciasidade 4.

Os valores possíveis de y são $[0, 1, 2, 4, 5, 6]$. Como há 6 valores possíveis de y , o procedimento deve retornar 6.

Restrições

- $1 \leq k \leq 60$
- $1 \leq q \leq 1000$
- $1 \leq x \leq 10^{18}$
- $0 \leq a[i] \leq 10^{18}$ (para todo $0 \leq i \leq k - 1$)
- Para cada chamada a `count_tastiness`, a soma dos valores de deliciasidade para todos

os biscoitos na despensa não excede 10^{18} .

Subtarefas

1. (9 pontos) $q \leq 10$, e para cada chamada a `count_tastiness`, a soma dos valores de deliciosidade para todos os biscoitos na despensa não excede 100 000.
2. (12 pontos) $x = 1$, $q \leq 10$
3. (21 pontos) $x \leq 10\,000$, $q \leq 10$
4. (35 pontos) O valor correto para o valor de retorno de cada chamada a `count_tastiness` não excede 200 000.
5. (23 pontos) Nenhuma restrição adicional.

Corretor exemplo

O corretor exemplo lê da entrada no seguinte formato. A primeira linha contém um inteiro q . Após, seguem q pares de linhas, sendo que cada par descreve um único cenário no seguinte formato:

- linha 1: k x
- linha 2: $a[0]$ $a[1]$ \dots $a[k - 1]$

A saída do corretor exemplo é no seguinte formato:

- linha i ($1 \leq i \leq q$): valor de retorno de `count_tastiness` para o i -ésimo cenário da entrada.