

Saquinhas de Biscoitos (biscuits)

A Tia Khong está a organizar uma competição com x participantes e quer dar a cada participante um **saquinho de biscoitos**. Existem k tipos de biscoitos diferentes, numerados de 0 a $k - 1$. Cada biscoito do tipo i ($0 \leq i \leq k - 1$) tem um **valor de sabor** igual a 2^i . A Tia Khong tem $a[i]$ (possivelmente zero) biscoitos do tipo i na despensa.

Cada um dos saquinhos da Tia Khong irá conter zero ou mais biscoitos de cada tipo. O número total de biscoitos do tipo i em todos os saquinhos não pode exceder $a[i]$. Chamamos à soma dos valores de sabor de todos os biscoitos de um saquinho o **sabor total** do saquinho.

Ajuda a Tia Khong a determinar quantos valores y diferentes existem tais que é possível formar x saquinhos de biscoitos, tendo cada um deles um sabor total igual a y .

Detalhes de Implementação

Deves implementar a seguinte função:

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

- x : o número de saquinhos de biscoitos a formar.
- a : um array de tamanho k . Para cada $0 \leq i \leq k - 1$, $a[i]$ indica o número de biscoitos do tipo i que há na despensa.
- A função deve retornar a quantidade de valores y diferentes tais que a Tia consegue formar x saquinhos de biscoitos, tendo cada um deles um sabor total igual a y .
- A função é chamada um total de q vezes (ver nas secções das Restrições e das Subtarefas os valores que q pode assumir). Cada uma dessas chamadas deve ser encarada como um cenário diferente.

Exemplos

Exemplo 1

Considera a seguinte chamada:

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

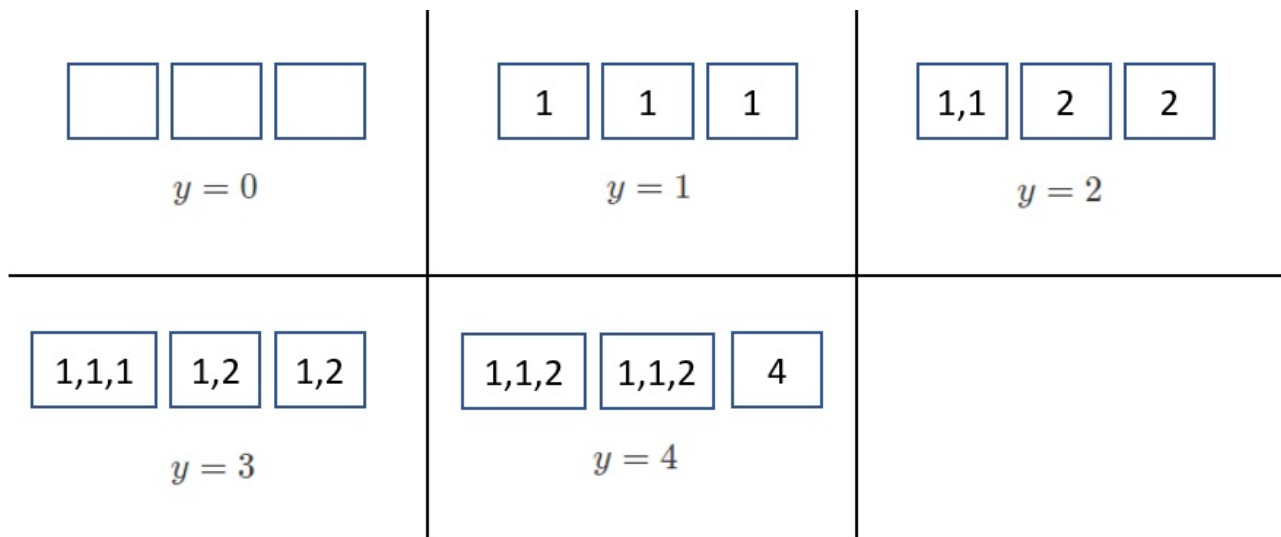
Isto significa que a Tia pretende formar 3 saquinhos e existem 3 tipos de biscoitos na despensa:

- 5 biscoitos do tipo 0, cada um tendo um valor de sabor igual a 1,
- 2 biscoitos do tipo 1, cada um tendo um valor de sabor igual a 2,
- 1 biscoito do tipo 2, tendo um valor de sabor igual a 4.

Os valores possíveis para y são $[0, 1, 2, 3, 4]$. Por exemplo, para formar 3 saquinhos com sabor total 3, a Tia pode formar:

- um saquinho contendo três biscoitos do tipo 0 e
- dois saquinhos, cada um contendo um biscoito do tipo 0 e um biscoito do tipo 1.

Uma vez que existem 5 valores possíveis para y , a função deve retornar 5.



Exemplo 2

Considera a seguinte chamada:

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

Isto significa que a Tia pretende formar 2 saquinhos e existem 3 tipos de biscoitos na despensa:

- 2 biscoitos do tipo 0, cada um tendo um valor de sabor igual a 1,
- 1 biscoito do tipo 1, tendo um valor de sabor igual a 2,
- 2 biscoitos do tipo 2, cada um tendo um valor de sabor igual a 4.

Os valores possíveis para y são $[0, 1, 2, 4, 5, 6]$. Uma vez que existem 6 valores possíveis para y , a função 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 o $0 \leq i \leq k - 1$)
- Em qualquer chamada a `count_tastiness`, a soma dos valores de sabor de todos os biscoitos na despensa não ultrapassa 10^{18} .

Subtarefas

1. (9 pontos) $q \leq 10$ e, em qualquer chamada a `count_tastiness`, a soma dos valores de sabor de todos os biscoitos na despensa não ultrapassa 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 de retorno correto de qualquer chamada a `count_tastiness` não ultrapassa 200 000.
5. (23 pontos) Sem restrições adicionais.

Avaliador exemplo

O avaliador exemplo lê o input no seguinte formato. A primeira linha contém um inteiro q . Seguem-se q pares de linhas, cada par descrevendo um único cenário no seguinte formato:

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

O avaliador exemplo escreve o output com o seguinte formato:

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