

# Складання печива (biscuits)

Онті Конг організує змагання для x участників та хоче дати кожному участнику **кошик печива**. Є k різних видів печива, пронумерованих від 0 до k-1. Кожна печенька типу i (  $0 \le i \le k-1$ ) має **рівень смакоти**  $2^i$ . Онті Конг має a[i] (можливо нуль) печеньок типу i у своїй коморі.

Кожен з кошиків Онті Конг буде містити нуль або більше печеньок кожного типу. Загальна кількість печива типу i у всіх кошиках має не перевищувати a[i]. Сума рівнів смакоти усіх печеньок у кошику називається **загальною смакотою** кошика.

Допоможіть Онті Конг з'ясувати скільки існує різних значень y, що можна скласти x кошиків печива, кожен з яких має загальну смакоту рівну y.

## Деталі реалізації

Ви маєте написати наступну процедуру:

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

- x: кількість кошиків печива, що треба скласти.
- a: масив довжини k. Для  $0 \le i \le k-1$ , a[i] позначає кількість печеньок типу i у коморі.
- Ця процедура має повертати кількість різних значень y таких, що Онті може скласти x кошиків печива, кожен з яких має загальну смакоту y.
- Ця процедура викликається q разів (дивіться розділи Обмеження та Підзадачі де вказано можливі значення q). Кожен з цих викликів має розглядатись як окремий сценарій.

## Приклади

#### Приклад 1

Розглянемо наступний виклик:

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

Це означає що Онті хоче скласти 3 кошики, та є 3 вида печива у коморі:

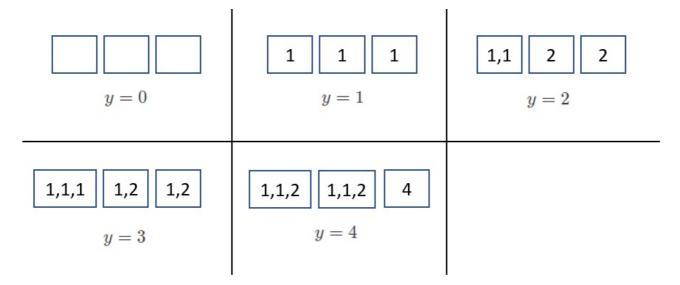
- 5 печеньок типу 0, кожна з яких має значення смакоти 1,
- 2 печеньки типу 1, кожна з яких має значення смакоти 2,

• 1 печенька типу 2, яка має значення смакоти 4.

Можливими значеннями  $y \in [0,1,2,3,4]$ . Наприклад, щоб скласти 3 кошика загальною смакотою 3, Онті може скласти:

- один кошик, що містить три печеньки типу 0, та
- два кошика, кожен з яких містить одну печеньку типу 0 та одну печеньку типу 1.

Оскільки є 5 можливих значень y, ця процедура має повернути 5.



#### Приклад 2

Розглянемо наступний виклик:

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

Це означає що Онті хоче скласти 2 кошики, та є 3 вида печива у коморі:

- ullet 2 печеньки типу 0, кожна з яких має значення смакоти 1,
- 1 печенька типу 1, яка має значення смакоти 2,
- 2 печеньки типу 2, кожна з яких має значення смакоти 4.

Можливими значеннями  $y \in [0,1,2,4,5,6]$ . Оскільки є 6 можливих значень y, ця процедура має повернути 6.

### Обмеження

- $1 \le k \le 60$
- $1 \le q \le 1000$
- $\bullet \ 1 \leq x \leq 10^{18}$
- ullet  $0 \leq a[i] \leq 10^{18}$  (для всіх  $0 \leq i \leq k-1$ )
- Для кожного виклику count tastiness сума значень смакоти всього печива у коморі не

перевищує  $10^{18}$ .

## Підзадачі

- 1. (9 балів)  $q \leq 10$ , та для кожного виклику count\_tastiness сума значень смакоти всього печива у коморі не перевищує 100~000.
- 2. (12 балів)  $x=1,\,q\le 10$
- 3. (21 бал)  $x \le 10~000$ ,  $q \le 10$
- 4. (35 балів) Правильне значення, що має повернути кожен виклик  $count\_tastiness$ , не перевищує  $200\ 000$ .
- 5. (23 бали) Без додаткових обмежень.

## Приклад модуля перевірки

Приклад модуля перевірки читає вхідні дані у наступному форматі. Перший рядок містить ціле число q. Після цього слідують q пар рядків, кожна з яких описує один сценарій у наступному форматі:

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

Вивід приклада модуля перевірки має наступний формат:

ullet рядок i ( $1 \leq i \leq q$ ): значення, що повернула <code>count\_tastiness</code> для i-го сценарію на вході.