



# Cepumu iepakošana (biscuits)

Tante Konga organizē sacensības  $x$  dalībniekiem un vēlas katram dalībniekam iedot **somu ar cepumiem**. Ir  $k$  dažādi cepumi veidi, kas numurēti no 0 līdz  $k - 1$ . Katram  $i$ -tā ( $0 \leq i \leq k - 1$ ) veida cepumam ir **garšīguma vērtība**  $2^i$ . Tantei Kongai skapī ir  $a[i]$  (iespējams nulle)  $i$ -tā veida cepumi.

Katrā tantes Kongas somā būs neviens, viens vai vairāki katra veida cepumi.  $i$ -tā veida cepumu skaits visās somās kopā nedrīkst pārsniegt  $a[i]$ . Visu cepumu garšīgumu vērtību summa vienā somā tiek saukta par somas cepumu **kopējo garšīgumu**.

Palīdzi tantei Kongai noskaidrot, cik daudz dažādu  $y$  vērtību pastāv, lai ir iespējams sapakot  $x$  somas tā, lai katrā somā cepumu kopējais garšīgums būtu  $y$ .

## Implementācijas detaļas

Tev ir jāimplementē šāda procedūra:

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

- $x$ : somu skaits, kurās jāiepako cepumi.
- $a$ :  $k$  elementu masīvs. Visiem  $i$  ( $0 \leq i \leq k - 1$ )  $a[i]$  apzīmē  $i$ -tā veida cepumu skaitu skapī.
- Procedūrai ir jāatgriež dažādo  $y$  vērtību skaits, lai tante Konga var iepakot cepumus  $x$  somās, lai katrā somā cepumu kopējais garšīgums būtu  $y$ .
- Procedūra tiek izsaukta pavisam  $q$  reizes (skatīt lerobežojumu un Apakšuzdevumu sadaļas). Katrs šis izsaukums ir jāapstrādā kā atsevišķs gadījums.

## Piemēri

### 1. piemērs

Aplūkosim šādu izsaukumu:

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

Tas nozīmē, ka tante Konga vēlas sapakot 3 somas un ka skapī ir 3 veidu cepumi:

- Pieci 0. veida cepumi, kuriem katram garšīguma vērtība ir 1,
- Divi 1. veida cepumi, kuriem katram garšīguma vērtība ir 2,

- Viens 2. veida cepums, kuram garšīguma vērtība ir 4.

Iespējamās  $y$  vērtības ir  $[0, 1, 2, 3, 4]$ . Piemēram, lai sarakot 3 somas ar cepumu kopējo garšīgumu 3, tante Konga var iepakot:

- vienu somu, kurā ir trīs 0. veida cepumi, un
- divas somas, kurās katrā ir viens 0. veida cepums un viens 1. veida cepums.

Procedūrai ir jāatgriež 5, jo  $y$  var būt 5 dažādas vērtības.

<div> <div></div> <div></div> <div></div> </div> $y = 0$	<div> <div>1</div> <div>1</div> <div>1</div> </div> $y = 1$	<div> <div>1,1</div> <div>2</div> <div>2</div> </div> $y = 2$
<div> <div>1,1,1</div> <div>1,2</div> <div>1,2</div> </div> $y = 3$	<div> <div>1,1,2</div> <div>1,1,2</div> <div>4</div> </div> $y = 4$	

## 2. piemērs

Aplūkosim šādu izsaukumu:

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

Tas nozīmē, ka tante Konga vēlas sarakot 2 somas un ka skapī ir 3 veidu cepumi:

- Divi 0. veida cepumi, kuriem katram garšīguma vērtība ir 1,
- Viens 1. veida cepums, kuram garšīguma vērtība ir 2,
- Divi 2. veida cepumi, kuriem katram garšīguma vērtība ir 4.

Iespējamās  $y$  vērtības ir  $[0, 1, 2, 4, 5, 6]$ . Procedūrai ir jāatgriež 6, jo  $y$  var būt 6 dažādas vērtības.

## Ierobežojumi

- $1 \leq k \leq 60$
- $1 \leq q \leq 1000$
- $1 \leq x \leq 10^{18}$
- $0 \leq a[i] \leq 10^{18}$  (visiem  $0 \leq i \leq k - 1$ )
- Katram `count_tastiness` izsaukumam visu skapī esošo cepumu garšīguma vērtību summa nepārsniedz  $10^{18}$ .

## Apakšuzdevumi

1. (9 punkti)  $q \leq 10$ , un katram `count_tastiness` izsaukumam visu skapī esošo cepumu garšīguma vērtību summa nepārsniedz 100 000.
2. (12 punkti)  $x = 1, q \leq 10$
3. (21 punkts)  $x \leq 10\,000, q \leq 10$
4. (35 punkti) Katra `count_tastiness` izsaukuma pareiza atgriežamā vērtība nepārsniedz 200 000.
5. (23 punkti) Bez papildu ierobežojumiem.

## Paraugvērtētājs

Paraugvērtētājs nolasa ievaddatus šādā formātā. Pirmā rinda satur veselu skaitli  $q$ . Pēc tam ir  $q$  rindu pāri, kuri katrs apraksta vienu gadījumu šādā formātā:

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

Paraugvērtētājs izvada datus šādā formātā:

- $i$ -tā rinda ( $1 \leq i \leq q$ ):  $i$ -tajam ievaddatu gadījumam `count_tastiness` atgrieztā vērtība.