

## Detekcija molekula

Petar je zaposlen u kompaniji koja proizvodi mašine za detekciju molekula. Svaki molekul ima cjelobrojnu težinu. Mašina ima *opseg detekcije*  $[l, u]$ , gdje su  $l$  i  $u$  cijeli brojevi. Mašina može detektovati skup molekula ako i samo ako skup sadrži podskup molekula čija je ukupna težina u opsegu detekcije mašine.

Formalno, posmatrajmo  $n$  molekula sa težinama koje su pozitivni cijeli brojevi.

$w_0, \dots, w_{n-1}$ . Mašina će uspješno obaviti detekciju ako postoji skup različitih indeksa  $I = \{i_1, \dots, i_m\}$  takav da  $l \leq w_{i_1} + \dots + w_{i_m} \leq u$ .

Mašina je konstruisana tako da je opseg između  $l$  i  $u$  garantovano veći ili jednak od opsega između najlakšeg i najtežeg molekula. Formalno,  $u - l \geq w_{max} - w_{min}$ , gdje je  $w_{max} = \max(w_0, \dots, w_{n-1})$  i  $w_{min} = \min(w_0, \dots, w_{n-1})$ .

Vaš zadatak je da napišete program koji će ili pronaći bilo koji podskup molekula čija je ukupna težina u opsegu detekcije mašine ili utvrditi da takav podskup ne postoji.

### Detalji implementacije

Potrebno je da implementirate funkciju (metod):

- `int[] solve(int l, int u, int[] w)`
  - $l$  i  $u$ : krajnje tačke opsega detekcije,
  - $w$ : težine molekula.
  - ako traženi podskup postoji, funkcija vraća niz indeksa molekula koji formiraju bilo koji takav podskup. Ako postoji više rješenja, vratiti bilo koje od njih.
  - ako traženi podskup ne postoji, funkcija vraća prazan niz.

Za jezik C signatura (zaglavlje) funkcije je:

- `int solve(int l, int u, int[] w, int n, int[] result)`
  - $n$ : broj elementa u nizu  $w$  (tj. broj molekula),
  - ostali parametri su isti kao u gornjoj funkciji (metodu).
  - umjesto da vrati niz od  $m$  indeksa (kao gore), funkcija upisuje indekse u prvih  $m$  elemenata niza `result` i vraća  $m$ .
  - ako traženi podskup ne postoji, funkcija ne upisuje ništa u niz `result` i vraća `0`.

Vaš program može upisivati indekse u vraćeni niz (ili u niz `result` za jezik C) u bilo kom redosljedu.

Molimo vas da koristite date templejt-fajlove za odgovarajući programski jezik.

## Primjeri

### Primjer 1

`solve(15, 17, [6, 8, 8, 7])`

U ovom primjeru data su četiri molekula sa težinama 6, 8, 8 i 7. Mašina može detektovati molekule čija je ukupna težina između 15 i 17, uključivo. Primjetite da važi  $17 - 15 \geq 8 - 6$ . Ukupna težina molekula 1 i 3 je  $w_1 + w_3 = 8 + 7 = 15$ , pa funkcija može vratiti `[1, 3]`. Druga rješenja su `[1, 2]` ( $w_1 + w_2 = 8 + 8 = 16$ ) i `[2, 3]` ( $w_2 + w_3 = 8 + 7 = 15$ ).

### Primjer 2

`solve(14, 15, [5, 5, 6, 6])`

U ovom primjeru data su četiri molekula sa težinama 5, 5, 6 i 6, i tražimo podskup čija je ukupna težina između 14 i 15, uključivo. Primjetite da opet važi  $15 - 14 \geq 6 - 5$ . Ne postoji podskup molekula čija je ukupna težina između 14 i 15 pa funkcija vraća prazan niz.

### Primjer 3

`solve(10, 20, [15, 17, 16, 18])`

U ovom primjeru data su četiri molekula sa težinama 15, 17, 16 i 18, i tražimo podskup čija je ukupna težina između 10 i 20, uključivo. Kao i u prethodnim primjerima, važi  $20 - 10 \geq 18 - 15$ . Bilo koji podskup koji sadrži tačno jedan element zadovoljava uslove, pa su tačna rješenja: `[0]`, `[1]`, `[2]` i `[3]`.

## Podzadaci

- (9 bodova):  $1 \leq n \leq 100$ ,  $1 \leq w_i \leq 100$ ,  $1 \leq u, l \leq 1000$ , svi  $w_i$  su međusobno jednaki.
- (10 bodova):  $1 \leq n \leq 100$ ,  $1 \leq w_i, u, l \leq 1000$  i  $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$ .
- (12 bodova):  $1 \leq n \leq 100$  i  $1 \leq w_i, u, l \leq 1000$ .
- (15 bodova):  $1 \leq n \leq 10000$  i  $1 \leq w_i, u, l \leq 10000$ .
- (23 boda):  $1 \leq n \leq 10000$  i  $1 \leq w_i, u, l \leq 500000$ .
- (31 bod):  $1 \leq n \leq 200000$  i  $1 \leq w_i, u, l < 2^{31}$ .

## Sample grader

Sistem za ocjenjivanje učitava ulazne podatke u sljedećem formatu:

- red 1: cijeli brojevi  $n$ ,  $l$ ,  $u$ .
- red 2:  $n$  cijelih brojeva:  $w_0, \dots, w_{n-1}$ .