

## Molekulide tuvastamine

Petr töötab ettevõttes, mis toodab molekulide tuvastamise masinaid. Iga molekuli mass on positiivne täisarv. Masinal on *tuvastamise lõik*  $[l, u]$ , kus  $l$  ja  $u$  on positiivsed täisarvud. Masin suudab mingi molekulide hulga tuvastada parajasti siis, kui selle hulga mingi alamhulga masside summa on masina tuvastamise lõigus.

Formaalsemalt, olgu meil  $n$  molekuli massidega  $w_0, \dots, w_{n-1}$ . Tuvastamine õnnestub, kui leidub selline paarikaupa erinevate indeksite hulk  $I = \{i_1, \dots, i_m\}$ , et  $l \leq w_{i_1} + \dots + w_{i_m} \leq u$ .

Masina iseärasuste tõttu võib eeldada, et  $l$  ja  $u$  vahe on alati vähemalt sama suur kui kergeima ja raskeima molekuli masside vahe. Formaalsemalt,  $u - l \geq w_{max} - w_{min}$ , kus  $w_{max} = \max(w_0, \dots, w_{n-1})$  ja  $w_{min} = \min(w_0, \dots, w_{n-1})$ .

Kirjutada programm, mis leiab molekulide hulga sellise alamhulga, mille summaarne mass jääb tuvastamise lõiku, või teeb kindlaks, et sellist alamhulka pole.

### Realisatsioon

Sinu lahendus peab realiseerima järgmise funktsiooni (meetodi):

- `int[] solve(int l, int u, int[] w)`
  - $l$  ja  $u$ : tuvastamise lõigu otspunktid;
  - $w$ : molekulide massid;
  - kui otsitav alamhulk leidub, peab funktsioon tagastama massiivi sellesse alamhulka kuuluvate elementide indeksitega; kui võimalikke vastuseid on mitu, tagastada ükskõik milline neist;
  - kui otsitavat alamhulka ei leidu, peab funktsioon tagastama tühja massiivi.

C keeles on funktsiooni liides natuke teistsugune:

- `int solve(int l, int u, int[] w, int n, int[] result)`
  - $n$ : massiivi  $w$  elementide arv (molekulide arv);
  - teised sisendparameetrid on samasugused nagu eelmisel juhul;
  - $m$  indeksiga massiivi tagastamise asemel peab funktsioon salvestama indeksid massiivi `result` esimesse  $m$  elementi ja seejärel tagastama  $m$ ;
  - kui otsitavat alamhulka ei leidu, peab funktsioon massiivi `result` puutumata jätma ja tagastama `0`.

Indeksid võivad tagastatavas massiivis (või C keeles massiivis `result`) olla mistahes

järjekorras.

Vaata ka näitekoodi failides olevat keelespetsiifilist lisainfot.

## Näited

### Näide 1

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

Selles näites on neli molekuli massidega 6, 8, 8 ja 7. Masin suudab tuvastada molekulide alamhulki kogumassiga 15 kuni 17 (kaasa arvatud). Pane tähele, et  $17 - 15 \geq 8 - 6$ . Molekulide 1 ja 3 masside summa on  $w_1 + w_3 = 8 + 7 = 15$ , seega võiks funktsioon tagastada `[1, 3]`. Teised võimalikud õiged vastused on `[1, 2]` ( $w_1 + w_2 = 8 + 8 = 16$ ) ja `[2, 3]` ( $w_2 + w_3 = 8 + 7 = 15$ ).

### Näide 2

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

Selles näites on neli molekuli massidega 5, 5, 6 ja 6 ning me otsime alamhulka summaarse massiga 14 kuni 15 (kaasa arvatud). Jällegi,  $15 - 14 \geq 6 - 5$ . Kuna ei ole ühtki alamhulka, mille masside summa oleks 14 või 15, peab funktsioon tagastama tühja massiivi.

### Näide 3

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

Selles näites on neli molekuli massidega 15, 17, 16 ja 18 ning me otsime alamhulka summaarse massiga 10 kuni 20 (kaasa arvatud). Jällegi,  $20 - 10 \geq 18 - 15$ . Iga üheelemendilise alamhulga mass jääb 10 ja 20 vahele, seega on võimalikud vastused `[0]`, `[1]`, `[2]` ja `[3]`.

## Alamülesanded

- (9 punkti):  $1 \leq n \leq 100$ ;  $1 \leq w_i \leq 100$ ;  $1 \leq u, l \leq 1000$  ja kõik  $w_i$  on võrdsed.
- (10 punkti):  $1 \leq n \leq 100$ ;  $1 \leq w_i, u, l \leq 1000$  ja  $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$ .
- (12 punkti):  $1 \leq n \leq 100$  ja  $1 \leq w_i, u, l \leq 1000$ .
- (15 punkti):  $1 \leq n \leq 10\,000$  ja  $1 \leq w_i, u, l \leq 10\,000$ .
- (23 punkti):  $1 \leq n \leq 10\,000$  ja  $1 \leq w_i, u, l \leq 500\,000$ .
- (31 punkti):  $1 \leq n \leq 200\,000$  ja  $1 \leq w_i, u, l < 2^{31}$ .

## Näitekood

Näitekood loeb sisendi järgmisel kujul:

- Rida 1: kolm täisarvu  $n, l, u$ .
- Rida 2:  $n$  täisarvu  $w_0, \dots, w_{n-1}$ .