

Molekulu noteikšana

Petrs strādā kompānijā, kas būvē mašīnas molekulu noteikšanai. Katrai molekulai ir svars, kas izsakāms kā naturāls skaitlis. Mašīnai ir *noteikšanas diapazons* $[l, u]$, kur l un u ir naturāli skaitļi. Mašīna var noteikt molekulu kopu tad un tikai tad, ja tā satur molekulu apakškopu ar kopējo svaru, kas pieder mašīnas noteikšanas diapazonam.

Formāli, ja n molekulu svāri ir naturāli skaitļi w_0, \dots, w_{n-1} , tad noteikšana ir veiksmīga tad, ja ir tāda dažādu indeksu kopa $I = \{i_1, \dots, i_m\}$, ka $l \leq w_{i_1} + \dots + w_{i_m} \leq u$.

Starpība starp l un u garantēti ir lielāka vai vienāda ar smagākās un vieglākās molekulas svaru starpību. Formāli, $u - l \geq w_{max} - w_{min}$, kur $w_{max} = \max(w_0, \dots, w_{n-1})$ un $w_{min} = \min(w_0, \dots, w_{n-1})$.

Jūsu uzdevums ir uzrakstīt programmu, kura vai nu atrod vienu molekulu apakškopu ar kopējo svaru noteikšanas diapazonā, vai arī noskaidro, ka tādas apakškopas nav.

Implementācijas detaļas

Jums ir jāimplementē viena funkcija (metode):

- `int[] solve(int l, int u, int[] w)`
 - l un u : noteikšanas diapazona galapunkti,
 - w : molekulu svāri.
 - ja vajadzīgā apakškopa eksistē, funkcijai ir jāatgriež to molekulu, kuras formē šo apakškopu, indeksu masīvu. Ja ir vairākas derīgas apakškopas, izvadiet informāciju par jebkuru no tām.
 - ja vajadzīgā apakškopa neeksistē, funkcijai ir jāatgriež tukšs masīvs.

C valodai funkcijas signatūra ir mazliet atšķirīga:

- `int solve(int l, int u, int[] w, int n, int[] result)`
 - n : elementu skaits masīvā w (t.i., molekulu skaits),
 - citi parametri ir tādi paši kā iepriekšējās funkcijas aprakstā.
 - indeksu masīva atgriešanas vietā (kā iepriekšējā funkcijā), funkcijai ir jāieraksta indeksi pirmajās m masīva `result` šūnās un jāatgriež m .
 - ja vajadzīgā apakškopa neeksistē, funkcijai masīvā `result` nekas nav jāraksta un ir jāatgriež `0`.

Atgriežamajā masīvā (vai masīvā `result` valodā C) indeksus drīkst izvadīt jebkurā

secībā.

Implementācijas detaļām lūdzu izmantotiet piedāvātos šablona failus jūsu izmantotajā programmēšanas valodā.

Piemēri

1. piemērs

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

šajā piemērā ir četras molekulas ar svāriem 6, 8, 8 and 7. Mašīna var noteikt molekulu apakškopas ar kopējo svaru starp 15 un 17 ieskaitot. Ievērojiet, ka $17 - 15 \geq 8 - 6$.

Molekulu 1 un 3 kopējais svārs ir $w_1 + w_3 = 8 + 7 = 15$, tāpēc funkcija var atgriezt `[1, 3]`. Citas derīgas atbildes ir `[1, 2]` ($w_1 + w_2 = 8 + 8 = 16$) un `[2, 3]` ($w_2 + w_3 = 8 + 7 = 15$).

2. piemērs

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

šajā piemērā ir četras molekulas ar svāriem 5, 5, 6 and 6, un mēs meklējam molekulu apakškopu ar svāriem starp 14 un 15 ieskaitot. Ievērojiet, ka $15 - 14 \geq 6 - 5$.

Molekulu apakškopa ar kopējo svaru starp 14 and 15 neeksistē, tāpēc funkcijai ir jāatgriež tukšs masīvs.

3. piemērs

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

šajā piemērā ir četras molekulas ar svāriem 15, 17, 16 and 18, un mēs meklējam molekulu apakškopu ar svāriem starp 10 un 20 ieskaitot. Ievērojiet ka

$20 - 10 \geq 18 - 15$. Jebkurai apakškopai no viena elementa kopējais molekulu svārs ir starp 10 un 20, tāpēc pareizas atbildes ir: `[0]`, `[1]`, `[2]` un `[3]`.

Apakšuzdevumi

- (9 punkti): $1 \leq n \leq 100$, $1 \leq w_i \leq 100$, $1 \leq u, l \leq 1000$, visi w_i ir vienādi.
- (10 punkti): $1 \leq n \leq 100$, $1 \leq w_i, u, l \leq 1000$ un $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$.
- (12 punkti): $1 \leq n \leq 100$ un $1 \leq w_i, u, l \leq 1000$.
- (15 punkti): $1 \leq n \leq 10000$ un $1 \leq w_i, u, l \leq 10000$.
- (23 punkti): $1 \leq n \leq 10000$ un $1 \leq w_i, u, l \leq 500000$
- (31 punkti): $1 \leq n \leq 200000$ un $1 \leq w_i, u, l < 2^{31}$.

Piemēru vērtētājs

Piemēru vērtētājs lasa ievaddatus sekojošā formātā:

- 1. rinda: naturāli skaitļi n , l , u .
- 2. rinda: n naturāli skaitļi w_0, \dots, w_{n-1} .

