



მოლეკულების აღმოჩენა

პეტრე მუშაობს კომპანიაში, რომელმაც შექმნა მოლეკულების აღმოსაჩენი მანქანა. თითოეული მოლეკულის წონა მთელი დადებითი რიცხვით გამოისახება. მანქანას გააჩნია ე.წ. აღმოჩენის დიაპაზონი $[l, u]$, სადაც l და u მთელი დადებითი რიცხვებია. მას შეუძლია აღმოაჩინოს მოლეკულათა სიმრავლე მაშინ და მხოლოდ მაშინ, თუ ეს სიმრავლე შეიცავს ისეთ მოლეკულათა ქვესიმრავლეს, რომელთა ჯამური წონა ეკუთვნის მანქანის აღმოჩენის დიაპაზონს.

ფორმალურად, განვიხილოთ n რაოდენობის მოლეკულა წონებით w_0, \dots, w_{n-1} . აღმოჩენა წარმატებულად ითვლება, თუ არსებობს განსხვავებულ ინდექსთა ისეთი $I = i_1, \dots, i_m$ მიმდევრობა, რომ $l \leq w_{i_1} + \dots + w_{i_m} \leq u$.

მანქანის სპეციფიკიდან გამომდინარე, u -სა და l -ს შორის სხვაობა გარანტირებულად მეტი ან ტოლია ყველაზე მძიმე და ყველაზე მსუბუქი მოლეკულების წონების სხვაობაზე. ფორმალურად, $u - l \geq w_{max} - w_{min}$, სადაც $w_{max} = \max(w_0, \dots, w_{n-1})$ და $w_{min} = \min(w_0, \dots, w_{n-1})$.

თქვენ უნდა დაწეროთ პროგრამა, რომელიც ან იპოვის მოლეკულათა რომელიმე ერთ-ერთ ისეთ ქვესიმრავლეს, რომელშიც შემავალი მოლეკულების ჯამური წონა ეკუთვნის აღმოჩენის დიაპაზონს, ან დაადგენს, რომ ასეთი ქვესიმრავლე არ არსებობს.

იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ ერთი ფუნქციის იმპლემენტაცია (მეთოდი):

- `int[] solve(int l, int u, int[] w)`
 - l და u : აღმოჩენის დიაპაზონის განაპირა წერტილებია,
 - w : მოლეკულათა წონები.
 - თუ მოთხოვნილი ქვესიმრავლე არსებობს, ფუნქციამ უნდა დააბრუნოს იმ მოლეკულათა ინდექსების მასივი, რომლებიც ერთ-ერთ ასეთ ქვესიმრავლეს ადგენენ. თუ არსებობს რამდენიმე ამონახსნი, დააბრუნეთ ერთი რომელიმე მათგანი.
 - თუ მოთხოვნილი ქვესიმრავლე არ არსებობს, ფუნქციამ უნდა დააბრუნოს ცარიელი მასივი.

For the C language the function signature is slightly different:

- `int solve(int l, int u, int[] w, int n, int[] result)`
 - n : the number of elements in w (i.e., the number of molecules),
 - the other parameters are the same as above.
 - instead of returning an array of m indices (as above), the function should write the indices to the first m cells of array `result` and then return m .

- if the required subset does not exist, the function should not write anything to the **result** array and it should return **0**.

Please use the provided template files for details of implementation in your programming language.

მაგალითები

მაგალითი 1

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

ამ მაგალითში ჩვენ გვაქვს ოთხი მოლეკულა წონებით: 6, 8, 8 და 7. მანქანას შეუძლია მოლეკულათა ქვესიმრავლეების განსაზღვრა ჯამური წონით 15-სა და 17-ს შორის (საზღვრების ჩათვლით). შევნიშნოთ, რომ $17 - 15 \geq 8 - 6$. 1-ლი და მე-3 მოლეკულების ჯამური წონა $w_1 + w_3 = 8 + 7 = 15$. ასე, რომ ფუნქციამ უნდა დააბრუნოს **[1, 3]**. სხვა შესაძლო სწორი ამონახსნები არის **[1, 2]** ($w_1 + w_2 = 8 + 8 = 16$) და **[2, 3]** ($w_2 + w_3 = 8 + 7 = 15$).

მაგალითი 2

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

ამ მაგალითში ჩვენ გვაქვს ოთხი მოლეკულა წონებით: 5, 5, 6, 6, და ჩვენ ვეძებთ მათ ისეთ ქვესიმრავლეს, რომელში შემავალი მოლეკულების ჯამური წონაც მოთავსებულია 14-სა და 15-ს შორის (საზღვრების ჩათვლით). ისევ შევნიშნოთ, რომ $15 - 14 \geq 6 - 5$. ამ შემთხვევაში მოლეკულათა ასეთი ქვესიმრავლე არ არსებობს, ამიტომ ფუნქციამ უნდა დააბრუნოს ცარიელი მასივი.

მაგალითი 3

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

ამ მაგალითში ჩვენ გვაქვს ოთხი მოლეკულა წონებით: 15, 17, 16, 18, და ჩვენ ვეძებთ მათ ისეთ ქვესიმრავლეს, რომელში შემავალი მოლეკულების ჯამური წონაც მოთავსებულია 10-სა და 20-ს შორის (საზღვრების ჩათვლით). ისევ შევნიშნოთ, რომ $20 - 10 \geq 18 - 15$. ნებისმიერ ერთელემენტიან ქვესიმრავლეში შემავალი მოლეკულის წონა მოთავსებულია 10-სა და 20-ს შორის. ასე, რომ შესაძლო სწორი ამონახსნებია: **[0]**, **[1]**, **[2]** და **[3]**.

ქვეამოცანები

- (9 ქულა): $1 \leq n \leq 100$, $1 \leq w_i \leq 100$, $1 \leq u, l \leq 1000$, ყველა w_i ტოლია.
- (10 ქულა): $1 \leq n \leq 100$, $1 \leq w_i, u, l \leq 1000$ და $\max(w_0, \dots, w_{n-1}) - \min(w_0, \dots, w_{n-1}) \leq 1$.
- (12 ქულა): $1 \leq n \leq 100$ და $1 \leq w_i, u, l \leq 1000$.
- (15 ქულა): $1 \leq n \leq 10000$ და $1 \leq w_i, u, l \leq 10000$.
- (23 ქულა): $1 \leq n \leq 10000$ და $1 \leq w_i, u, l \leq 500000$.
- (31 ქულა): $1 \leq n \leq 200000$ და $1 \leq w_i, u, l < 2^{31}$.

სანიმუშო გრაფერი

სანიმუშო გრადერი შეტანას კითხულობს შემდეგ ფორმატში:

- სტრიქონი 1: მთელი რიცხვები n , l , u .
- სტრიქონი 2: n რაოდენობის მთელი რიცხვი: w_0, \dots, w_{n-1} .