Εντοπισμός Μορίων

Ο Πέτρος εργάζεται για μια εταιρεία που έχει φτιάξει μια μηχανή εντοπισμού μορίων. Κάθε μόριο έχει ένα ακέραιο βάρος. Η μηχανή έχει ένα *διάστημα εντοπισμού* $[l, u]$, όπου τα l και u είναι ακέραιοι. Η μηχανή μπορεί να εντοπίσει ένα σύνολο από μόρια αν και μόνο αν αυτό το σύνολο περιέχει ένα υποσύνολο των μορίων με ολικό βάρος στο διάστημα εντοπισμού της μηχανής.

Τυπικά, έστω n μόρια με θετικά ακέραια βάρη $w_0, \ldots, w_{n-1}$. Ο εντοπισμός είναι επιτυχής αν υπάρχει ένα σύνολο διακριτών δεικτών $I = {i_1, \ldots , i_m}$ τέτοιο ώστε $l \le w_{i_1} + \ldots w_{i_m} \le u$.

Λόγω απαιτήσεων της μηχανής, το διάστημα μεταξύ l και u είναι σίγουρα μεγαλύτερο ή ίσο του διαστήματος βάρους μεταξύ του βαρύτερου και του ελαφρύτερου μορίου. Τυπικά, $u - l \ge w_{max} - w_{min}$, όπου $w_{max}=\max(w_0, \ldots , w_{n-1})$ και $w_{min}=\min(w_0, \ldots , w_{n-1})$.

Θα γράψετε ένα πρόγραμμα που είτε θα βρίσκει οποιοδήποτε υποσύνολο μορίων με ολικό βάρος στο διάστημα εντοπισμού, ή θα αποφαίνεται ότι δεν υπάρχει τέτοιο υποσύνολο.

Λεπτομέρειες υλοποίησης

Θα πρέπεινα υλοποιήσετε μια συνάρτηση (function, method):

- `int[] solve(int l, int u, int[] w)`
  - `l` και `u`: τα άκρα του διαστήματος εντοπισμού,
  - `w`: τα βάρη των μορίων.
  - αν το ζητούμενο υποσύνολο υπάρχει, η συνάρτηση θα πρέπει να επιστρέφει ένα πίνακα (array) από δείκτες μορίων που αποτελούν ένα τέτοιο υποσύνολο. Αν υπάρχουν πολλές διαφορετικές απαντήσεις, να επιστρέφει οποιαδήποτε απο αυτές.
  - αν το ζητούμενο υποσύνολο δεν υπάρχει, η συνάρτηση θα πρέπει να επιστρέφει ένα κενό πίνακα.

Για τη γλώσσα C η επικεφαλίδα της συνάρτησης είναι ελαφρώς διαφορετική:

- `int solve(int l, int u, int[] w, int n, int[] result)`
  - `n`: το πλήθος των στοιχείων του `w` (δηλαδή, το πλήθος των μορίων),
  - οι υπόλοιπες παράμετροι είναι ίδιες με τις παραπάνω.
  - αντί να επιστρέφει ένα πίνακα από m δείκτες (όπως παραπάνω), η συνάρτηση θα πρέπει να γράφει τους δείκτες στις πρώτες m θέσεις του πίνακα `result` και στη συνέχεια να επιστρέφει το m.
  - αν το ζητούμενο υποσύνολο δεν υπάρχει, η συνάρτηση δεν θα πρέπει να γράφει κάτι στον πίνακα `result` και θα πρέπει να επιστρέφει 0.

Το πρόγραμμά σας μπορεί να γράφει τους δείκτες στον πίνακα που επιστρέφει (ή στην C στον πίνακα `result`) με οπιαδήποτε σειρά.

Για λεπτομέρειες υλοποίησης στη γλώσσα προγραμματισμού σας χρησιμοποιήστε τα δοθέντα πρότυπα αρχείων.

Παραδείγματα

Παράδειγμα 1

`solve(15, 17, [6, 8, 8, 7])`
Σε αυτό το παράδειγμα έχουμε τέσσερα μόρια με βάρη 6, 8, 8 και 7. Η μηχανή μπορεί να εντοπίσει μόρια με συνολικό βάρος μεταξύ 15 και 17, συμπεριλαμβανομένων. Σημειώστε, ότι $17-15 \ge 8-6$. Το ολικό βάρος των μορίων 1 και 3 είναι $w_1 + w_3 = 8 + 7 = 15$, έτσι η συνάρτηση μπορεί να επιστρέψει `[1, 3]`. Άλλες πιθανές σωστές απαντήσεις είναι `[1, 2]` ($w_1 + w_2 = 8 + 8 = 16$) και `[2, 3]` ($w_2 + w_3 = 8 + 7 = 15$).

Παράδειγμα 2

`solve(14, 15, [5, 5, 6, 6])`
Στο παράδειγμα αυτό έχουμε τέσσερα μόρια με βάρη 5, 5, 6 και 6, και αναζητούμε ένα υποσύνολό τους με συνολικό βάρος μεταξύ 14 and 15, συμπεριλαβνομένων. Ξανά, σημειώστε ότι $15-14 \ge 6-5$. Δεν υπάρχει υποσύνολο μορίων με συνολικό βάρος μεταξύ 14 and 15 έτσι η συνάρτηση θα πρέπει να επιστρέψει ένα κενό πίνακα.

Παράδειγμα 3

`solve(10, 20, [15, 17, 16, 18])`
Στο παράδειγμα αυτό έχουμε τέσσερα μόρια με βάρη 15, 17, 16 και 18, και αναζητούμε ένα υποσύνολό τους με συνολικό βάρος μεταξύ 10 και 20, συμπεριλαμβανομένων. Ξανά, σημειώστε ότι $20-10 \ge 18-15$. Οποιοδήποτε υποσύνολο αποτελείται από ακριβώς ένα στοιχείο ικανοποιεί την απαίτηση, έτσι σωστές απαντήσεις είναι οι: `[0]`, `[1]`, `[2]` και `[3]`.

Υποπροβλήματα

1. (9 βαθμοί): $1 \le n \le 100$, $1 \le w_i \le 100$, $1 \le u, l \le 1000$, όλα τα w_i είναι ίσα.
2. (10 βαθμοί): $1 \le n \le 100$, $1 \le w_i, u, l \le 1000$ και $\max(w_0, \ldots , w_{n-1}) - \min(w_0, \ldots , w_{n-1}) \le 1$.
3. (12 βαθμοί): $1 \le n \le 100$ και $1 \le w_i,u,l \le 1000$.
4. (15 βαθμοί): $1 \le n \le 10\,000$ και $1 \le w_i,u,l \le 10\,000$.
5. (23 βαθμοί): $1 \le n \le 10\,000$ και $1 \le w_i,u,l \le 500\,000$.
6. (31 βαθμοί): $1 \le n \le 200\,000$ και $1 \le w_i,u,l < 2^{31}$.

Υπόδειγμα βαθμολογητή

Ο βαθμολογητής υπόδειγμα που δίδεται διαβάζει την είσοδο του με την εξής μορφή:

- γραμμή 1: ακέραιοι n, l, u.
- γραμμή 2: n ακέραιοι: $w_0, \ldots , w_{n-1}$.