

## Partition

Ο Temur και ο βοηθός του, ο Ulug'bek, ετοιμάζουν ένα μαγικό κόλπο για την εκπομπή *Uzbekistan Got Talent*. Το κόλπο σχετίζεται με την επίλυση του ακόλουθου **προβλήματος διαμέρισης** από τον Temur: να διαμεριστεί μια δεδομένη συλλογή θετικών ακεραίων σε  $K$  μη κενές ομάδες έτσι ώστε το άθροισμα των στοιχείων κάθε ομάδας να είναι το ίδιο. Με άλλα λόγια, κάθε ακεραίος αριθμός της συλλογής πρέπει να αντιστοιχισθεί σε ακριβώς μία από τις  $K$  ομάδες και το άθροισμα των στοιχείων κάθε ομάδας πρέπει να είναι ίσο. Για παράδειγμα, εάν η δεδομένη συλλογή είναι  $[2, 1, 6, 4, 5]$  και  $K = 3$ , μια έγκυρη διαμέριση θα ήταν  $[2, 4]$ ,  $[1, 5]$  και  $[6]$ . Σε αυτήν την περίπτωση, το άθροισμα των στοιχείων κάθε ομάδας είναι ακριβώς 6.

Το κόλπο εκτελείται ως εξής:

- Ο Ulug'bek και ο Temur μπαίνουν σε ξεχωριστά δωμάτια και δεν μπορούν να επικοινωνήσουν μεταξύ τους.
- Ένα μέλος της κριτικής επιτροπής δίνει στον Ulug'bek έναν πίνακα  $A$  αποτελούμενο από  $N$  θετικούς ακεραίους αριθμούς,  $A[0], A[1], \dots, A[N - 1]$ , όπου κάθε τιμή είναι μεταξύ 1 και  $M$ , συμπεριλαμβανομένων. Το μέλος της κριτικής επιτροπής δίνει επίσης στον Ulug'bek την τιμή του  $K$ .
- Ο Ulug'bek επιλέγει το πολύ  $K - 1$  (όχι απαραίτητα διακριτούς) ακεραίους αριθμούς για να **προστεθούν ως νέα στοιχεία** στον πίνακα  $A$ . Κάθε ακεραίος που προστίθεται πρέπει επίσης να είναι μεταξύ 1 και  $M$ , συμπεριλαμβανομένων.
- Η κριτική επιτροπή προσθέτει τους ακεραίους αριθμούς που επέλεξε ο Ulug'bek στον αρχικό πίνακα. Στη συνέχεια, ο πίνακας που προκύπτει **ταξινομείται σε αύξουσα σειρά** (όχι απαραίτητα γνησίως αύξουσα) και δίνεται στον Temur, μαζί με την τιμή του  $K$ .
- Ο Temur πρέπει στη συνέχεια να λύσει το πρόβλημα διαμέρισης για αυτόν τον ταξινομημένο πίνακα.

Ζητείται να σχεδιάσετε και να υλοποιήσετε μια στρατηγική για τον Temur και τον Ulug'bek. Μπορεί να αποδειχθεί ότι, με τους δεδομένους περιορισμούς, υπάρχει στρατηγική που επιτρέπει σε αυτούς τους δύο να επιλύσουν επιτυχώς το πρόβλημα της διαμέρισης, ανεξάρτητα από τον πίνακα  $A$  που δίνεται από την κριτική επιτροπή.

## Λεπτομέρειες υλοποίησης

Θα πρέπει να υλοποιήσετε δύο συναρτήσεις:

Η συνάρτηση που θα υλοποιήσετε για τον **Ulug'bek** είναι:

```
std::vector<int> add_numbers(std::vector<int> A, int K, int M)
```

- $A$ : ένας πίνακας μήκους  $N$  που αντιπροσωπεύει τον αρχικό πίνακα που δόθηκε στον Ulug'bek.
- $K$ : το ζητούμενο πλήθος ομάδων. Να σημειωθεί ότι ο Ulug'bek μπορεί να προσθέσει μέχρι  $K - 1$  ακραίους στον πίνακα.
- $M$ : η μέγιστη επιτρεπόμενη τιμή για κάθε αρχικό και νέο ακέραιο αριθμό.
- Αυτή η συνάρτηση καλείται ακριβώς μία φορά για κάθε περίπτωση ελέγχου.

Η συνάρτηση θα πρέπει να επιστρέφει έναν πίνακα  $C$  που περιέχει τους ακραίους αριθμούς που ο Ulug'bek θέλει να προσθέσει στον αρχικό πίνακα. Έστω  $S$  το μήκος του πίνακα  $C$ .

- Η τιμή του  $S$  πρέπει να είναι το πολύ  $K - 1$ .
- Κάθε στοιχείο του  $C$  πρέπει να κυμαίνεται μεταξύ 1 και  $M$ , συμπεριλαμβανομένων.

Η συνάρτηση που θα υλοποιήσετε για τον **Temur** είναι:

```
std::vector<int> find_partition(std::vector<int> B, int K)
```

- $B$ : ένας πίνακας μήκους  $N + S$  που περιέχει τόσο τους αρχικούς ακραίους από τον  $A$  όσο και τους ακραίους που πρόσθεσε ο Ulug'bek. Τα στοιχεία του πίνακα  $B$  ταξινομούνται σε (όχι απαραίτητα γνησίως) αύξουσα σειρά.
- $K$ : το ζητούμενο πλήθος ομάδων.
- Αυτή η συνάρτηση καλείται ακριβώς μία φορά για κάθε περίπτωση ελέγχου.

Η συνάρτηση θα πρέπει να επιστρέφει έναν πίνακα  $P$  που αντιπροσωπεύει την διαμέριση του  $B$  σε  $K$  ξένες μεταξύ τους ομάδες.

- Το μήκος του  $P$  θα πρέπει να είναι  $N + S$ .
- Για κάθε  $j$  τέτοιο ώστε  $0 \leq j < N + S$ , το στοιχείο  $P[j]$  δηλώνει την ομάδα στην οποία ανήκει  $B[j]$ .
- Οι ομάδες θα πρέπει να αριθμούνται από 0 έως και  $K - 1$ , που σημαίνει ότι για κάθε  $0 \leq j < N + S$  πρέπει να ισχύει  $0 \leq P[j] < K$ .
- Για κάθε  $i$  μεταξύ 0 και  $K - 1$ , συμπεριλαμβανομένων, πρέπει να υπάρχει τουλάχιστον ένα στοιχείο  $j$  (όπου  $0 \leq j < N + S$ ) τέτοιο ώστε  $P[j] = i$ .
- Το άθροισμα των στοιχείων που έχουν αντιστοιχισθεί σε κάθε μία από τις ομάδες  $K$  πρέπει να είναι το ίδιο.

Κατά τη βαθμολόγηση, ένα πρόγραμμα που καλεί τις παραπάνω συναρτήσεις εκτελείται **ακριβώς δύο φορές**.

- Κατά την πρώτη εκτέλεση του προγράμματος:
  - Η συνάρτηση `add_numbers` καλείται ακριβώς μία φορά.
  - Το σύστημα βαθμολόγησης επεξεργάζεται τον επιστρεφόμενο πίνακα για να υπολογίσει τον πίνακα  $B$ .

- Κατά τη δεύτερη εκτέλεση του προγράμματος:
  - Η συνάρτηση `find_partition` καλείται ακριβώς μία φορά.

## Περιορισμοί

- $3 \leq N \leq 100\,000$
- $2 \leq K \leq 100\,000$
- $K \leq N$
- $1 \leq M \leq 10^9$
- $1 \leq A[i] \leq M$  για κάθε  $i$  τέτοιο ώστε  $0 \leq i < N$ .

## Υποπροβλήματα

| Υποπρόβλημα | Βαθμολογία | Περαιτέρω περιορισμοί         |
|-------------|------------|-------------------------------|
| 1           | 5          | $N = 3$                       |
| 2           | 4          | $M = 1$                       |
| 3           | 7          | $M \leq 2$                    |
| 4           | 10         | $N \leq 10$                   |
| 5           | 7          | $K = 2$                       |
| 6           | 12         | $K \leq 3$                    |
| 7           | 17         | $K \leq 10$                   |
| 8           | 20         | $K \leq 100$                  |
| 9           | 18         | Χωρίς περαιτέρω περιορισμούς. |

## Παράδειγμα

Θεωρήστε την παρακάτω κλήση:

```
add_numbers([8, 2, 9, 6, 1, 5, 5], 3, 9)
```

Σε αυτό το παράδειγμα  $A = [8, 2, 9, 6, 1, 5, 5]$ . Θέλουμε να διαμερίσουμε τους αριθμούς σε  $K = 3$  ομάδες. Ο Ulug'bek μπορεί να προσθέσει αριθμούς μεταξύ 1 και  $M = 9$ .

Η διαδικασία μπορεί να επιστρέψει  $[5, 4]$ , που σημαίνει ότι ο Ulug'bek αποφασίζει να προσθέσει  $S = 2$  νέους ακέραιους αριθμούς: το 5 και το 4. Αυτό είναι έγκυρο καθώς και τα δύο στοιχεία είναι μεταξύ 1 και  $M = 9$ .

Στη συνέχεια, ο πίνακας που προκύπτει ταξινομείται και διαβιβάζεται στον Temur με την ακόλουθη κλήση:

```
find_partition([1, 2, 4, 5, 5, 5, 6, 8, 9], 3)
```

Μπορούμε να διαμερίσουμε τους ακραίους αυτού του πίνακα στις εξής τρεις ομάδες:  $[1, 5, 9]$ ,  $[2, 5, 8]$ , και  $[4, 5, 6]$ . Να σημειωθεί ότι το άθροισμα καθεμίας από αυτές τις ομάδες είναι ίσο με 15. Για να αναφερθεί αυτή η διαμέριση, η συνάρτηση θα πρέπει να επιστρέψει τον πίνακα  $[0, 1, 2, 0, 1, 2, 2, 1, 0]$ .

## Ενδεικτικός βαθμολογητής

Μορφή εισόδου:

```
N K M  
A[0] A[1] ... A[N-1]
```

Μορφή εξόδου:

Αφού ολοκληρωθεί η κλήση της συνάρτησης `add_numbers`, ο ενδεικτικός βαθμολογητής:

- Υπολογίζει τον ταξινομημένο πίνακα  $B$ .
- Τυπώνει τους πίνακες  $C$  και  $B$  στην ακόλουθη μορφή, **ακολουθούμενους από μια κενή γραμμή**:

```
S  
C[0] C[1] ... C[S-1]  
B[0] B[1] ... B[N+S-1]
```

Αφού ολοκληρωθεί η κλήση του `find_partition`, ο βαθμολογητής εκτυπώνει:

```
L  
P[0] P[1] ... P[L-1]
```

Εδώ,  $L$  είναι το μήκος του πίνακα  $P$  που επιστρέφεται από τη `find_partition`.