



Εισιτήρια για το καρναβάλι (tickets)

Ο Ringo είναι σε ένα καρναβάλι στην Σιγκαπούρη. Έχει στην τσάντα του μερικά εισιτήρια επιβράβευσης (κουπόνια), που θέλει να χρησιμοποιήσει. Κάθε εισιτήριο έχει ένα από n χρώματα και τυπωμένο πάνω του ένα μη αρνητικό ακέραιο. Οι ακέραιοι σε διαφορετικά εισιτήρια μπορεί να είναι ίδιοι. Για κάποιο λόγο, είναι σίγουρο ότι το n είναι **άρτιος** αριθμός.

Ο Ringo έχει στην τσάντα του m εισιτήρια από κάθε χρώμα, δηλαδή έχει συνολικά $n \cdot m$ εισιτήρια. Το j -οστό εισιτήριο χρώματος i έχει τυπωμένο έναν ακέραιο $x[i][j]$ ($0 \leq i \leq n - 1$ και $0 \leq j \leq m - 1$).

Το παιχνίδι επιβράβευσης παίζεται σε k γύρους, αριθμημένους από 0 έως $k - 1$. Κάθε γύρος παίζεται με την ακόλουθη σειρά:

- Ο Ringo διαλέγει από την τσάντα του ένα **σύνολο** από n εισιτήρια, ένα από κάθε χρώμα. Στη συνέχεια δίνει το σύνολο στον διαχειριστή του παιχνιδιού.
- Ο διαχειριστής του παιχνιδιού σημειώνει τους ακέραιους $a[0], a[1] \dots a[n - 1]$ που είναι τυπωμένοι στα εισιτήρια του συνόλου. Η σειρά αυτών των n ακεραίων δεν είναι σημαντική.
- Ο διαχειριστής του παιχνιδιού τραβάει στην τύχη μια ειδική κάρτα από ένα τυχερό κουτί και σημειώνει τον ακέραιο b που είναι τυπωμένος σε αυτήν την κάρτα.
- Ο διαχειριστής του παιχνιδιού υπολογίζει τις απόλυτες διαφορές μεταξύ $a[i]$ και b για κάθε i από 0 έως $n - 1$. Έστω S το άθροισμα όλων αυτών των απόλυτων διαφορών.
- Για τον γύρο αυτό, ο διαχειριστής του παιχνιδιού δίνει στον Ringo ένα βραβείο αξίας S .
- Τα εισιτήρια του συνόλου ακυρώνονται και δεν μπορούν να χρησιμοποιηθούν σε μελλοντικούς γύρους.

Τα υπόλοιπα εισιτήρια στην τσάντα του Ringo ακυρώνονται μετά από k γύρους.

Παρατηρώντας προσεκτικά, ο Ringo αντιλαμβάνεται ότι το βραβείο του παιχνιδιού είναι στημένο! Στην πραγματικότητα υπάρχει ένας εκτυπωτής στο τυχερό κουτί. Σε κάθε γύρο, ο διαχειριστής του παιχνιδιού βρίσκει έναν ακέραιο b που ελαχιστοποιεί την τιμή του βραβείου αυτού του γύρου. Η τιμή που επιλέγεται από τον διαχειριστή του παιχνιδιού τυπώνεται στη ειδική κάρτα αυτού του γύρου.

Έχοντας όλη αυτή την πληροφόρηση, ο Ringo θέλει να βρει πώς θα μοιράσει τα εισιτήρια στους γύρους του παιχνιδιού. Δηλαδή, θέλει να επιλέξει το σύνολο των εισιτηρίων που θα χρησιμοποιήσει σε κάθε γύρο για να μεγιστοποιήσει το συνολικό ποσό των βραβείων.

Λεπτομέρειες υλοποίησης

Θα πρέπει να υλοποιήσετε την παρακάτω συνάρτηση:

```
int64 find_maximum(int k, int[][] x)
```

- k : το πλήθος των γύρων.
- x : ένας πίνακας διαστάσεων $n \times m$ με τους ακέραιους κάθε εισιτηρίου. Τα εισιτήρια κάθε χρώματος είναι ταξινομημένα σε μη-φθίνουσα σειρά των ακεραίων τους.
- Αυτή η συνάρτηση καλείται ακριβώς μια φορά.
- Η συνάρτηση αυτή θα πρέπει να καλέσει ακριβώς μια φορά την `allocate_tickets` (βλ. παρακάτω), περιγράφοντας k σύνολα εισιτηρίων, ένα για κάθε γύρο. Η κατανομή θα πρέπει να μεγιστοποιεί την συνολική τιμή των βραβείων.
- Αυτή η συνάρτηση θα πρέπει να επιστρέφει την συνολική αξία των βραβείων.

Η συνάρτηση `allocate_tickets` ορίζεται ως εξής:

```
void allocate_tickets(int[][] s)
```

- s : ένας πίνακας διαστάσεων $n \times m$. Η τιμή του $s[i][j]$ θα πρέπει να είναι r αν το εισιτήριο j χρώματος i χρησιμοποιείται στο σύνολο του γύρου r του παιχνιδιού, ή -1 αν δεν χρησιμοποιείται καθόλου.
- Για κάθε $0 \leq i \leq n - 1$, μεταξύ των $s[i][0], s[i][1], \dots, s[i][m - 1]$ κάθε τιμή $0, 1, 2, \dots, k - 1$ πρέπει να εμφανίζεται μια φορά, και όλες οι άλλες τιμές να είναι -1 .
- Αν υπάρχουν πολλές διατάξεις που δίνουν την μέγιστη συνολική τιμή βραβείου, μπορείτε να αναφέρετε οποιαδήποτε από αυτές.

Παραδείγματα

Παράδειγμα 1

Έστω η παρακάτω κλήση:

```
find_maximum(2, [[0, 2, 5], [1, 1, 3]])
```

Αυτό σημαίνει ότι:

- υπάρχουν $k = 2$ γύροι
- οι ακέραιοι που είναι τυπωμένοι στα εισιτήρια χρώματος 0 είναι 0, 2 και 5, αντίστοιχα
- οι ακέραιοι που είναι τυπωμένοι στα εισιτήρια χρώματος 1 είναι 1, 1 και 3, αντίστοιχα.

Μια πιθανή διάταξη που δίνει την μέγιστη συνολική τιμή βραβείου είναι:

- Στο γύρο 0, ο Ringo επιλέγει το εισιτήριο 0 χρώματος 0 (με τον ακέραιο 0) και το εισιτήριο 2 χρώματος 1 (με τον ακέραιο 3). Η χαμηλότερη δυνατή τιμή βραβείου στον γύρο αυτό είναι 3. Δηλαδή, ο διαχειριστής του παιχνιδιού μπορεί να επιλέξει $b = 1$:
 $|1 - 0| + |1 - 3| = 1 + 2 = 3$.
- Στον γύρο 1, ο Ringo επιλέγει το εισιτήριο 2 χρώματος 0 (με τον ακέραιο 5) και το εισιτήριο 1

χρώματος 1 (με τον ακέραιο 1). Η χαμηλότερη πιθανή τιμή βραβείου σε αυτό τον γύρο είναι 4. Δηλαδή, ο διαχειριστής του παιχνιδιού μπορεί να επιλέξει $b = 3$:
 $|3 - 1| + |3 - 5| = 2 + 2 = 4$.

- Επομένως, η συνολική τιμή των βραβείων θα πρέπει να είναι $3 + 4 = 7$.

Για να βγει αυτό το αποτέλεσμα, θα πρέπει η διαδικασία `find_maximum` να εκτελέσει την ακόλουθη κλήση στην `allocate_tickets`:

- `allocate_tickets([[0, -1, 1], [-1, 1, 0]])`

Τέλος, η διαδικασία `find_maximum` θα πρέπει να επιστρέψει 7.

Παράδειγμα 2

Έστω η ακόλουθη κλήση:

```
find_maximum(1, [[5, 9], [1, 4], [3, 6], [2, 7]])
```

Αυτό σημαίνει ότι:

- υπάρχει μόνο ένας γύρος,
- οι ακέραιοι που είναι τυπωμένοι στα εισιτήρια χρώματος 0 είναι 5 και 9, αντίστοιχα
- οι ακέραιοι που είναι τυπωμένοι στα εισιτήρια χρώματος 1 είναι 1 και 4, αντίστοιχα
- οι ακέραιοι που είναι τυπωμένοι στα εισιτήρια χρώματος 2 είναι 3 και 6, αντίστοιχα
- οι ακέραιοι που είναι τυπωμένοι στα εισιτήρια χρώματος 3 είναι 2 και 7, αντίστοιχα.

Μια πιθανή διάταξη που δίνει την μέγιστη συνολική τιμή βραβείου είναι:

- Στο γύρο 0, ο Ringo επιλέγει το εισιτήριο 1 χρώματος 0 (με τον ακέραιο 9), το εισιτήριο 0 χρώματος 1 (με τον ακέραιο 1), το εισιτήριο 0 χρώματος 2 (με τον ακέραιο 3), το εισιτήριο 1 χρώματος 3 (με τον ακέραιο 7). Η χαμηλότερη πιθανή τιμή βραβείου σε αυτό τον γύρο είναι 12, όταν ο διαχειριστής του παιχνιδιού επιλέξει $b = 3$:
 $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

Για να αναφερθεί αυτή η λύση θα πρέπει η συνάρτηση `find_maximum` να εκτελέσει την παρακάτω κλήση της `allocate_tickets`:

- `allocate_tickets([[-1, 0], [0, -1], [0, -1], [-1, 0]])`

Τέλος, η συνάρτηση `find_maximum` θα πρέπει να επιστρέψει 12.

Περιορισμοί

- $2 \leq n \leq 1500$ και n είναι άρτιος.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (για κάθε $0 \leq i \leq n - 1$ και $0 \leq j \leq m - 1$)

- $x[i][j - 1] \leq x[i][j]$ (για κάθε $0 \leq i \leq n - 1$ και $1 \leq j \leq m - 1$)

Υποπροβλήματα

1. (11 βαθμοί) $m = 1$
2. (16 βαθμοί) $k = 1$
3. (14 βαθμοί) $0 \leq x[i][j] \leq 1$ (για κάθε $0 \leq i \leq n - 1$ και $0 \leq j \leq m - 1$)
4. (14 βαθμοί) $k = m$
5. (12 βαθμοί) $n, m \leq 80$
6. (23 βαθμοί) $n, m \leq 300$
7. (10 βαθμοί) Χωρίς επιπλέον περιορισμούς.

Υποδειγματικός βαθμολογητής

Ο υποδειγματικός βαθμολογητής διαβάζει την είσοδο ως εξής:

- γραμμή 1: $n \ m \ k$
- γραμμή $2 + i$ ($0 \leq i \leq n - 1$): $x[i][0] \ x[i][1] \ \dots \ x[i][m - 1]$

Ο υποδειγματικός βαθμολογητής τυπώνει την απάντηση ως εξής:

- γραμμή 1: η τιμή που επιστρέφει η `find_maximum`
- γραμμή $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$