The Big Prize — Το μεγάλο βραβείο

Το μεγάλο βραβείο είναι ένα διάσημο τηλεοπτικό show. Είσαι ο τυχερός διαγωνιζόμενος που έχει περάσει στον τελευταίο γύρο. Στέκεσαι μπροστά σε μια σειρά από n κουτιά, αριθμημένα από 0 έως $n-1$ από αριστερά προς τα δεξιά. Κάθε κουτί περιέχει ένα βραβείο που δεν μπορείς να δεις, μέχρι να ανοίξει το κουτί. Υπάρχουν $v \geq 2$ διαφορετικοί *τύποι* βραβείων. Οι τύποι των βραβείων είναι αριθμημένοι από 1 έως v, σε *φθίνουσα* σειρά αξίας.

Το βραβείο τύπου 1 είναι το πιο ακριβό από όλα: ένα διαμάντι. Υπάρχει ακριβώς ένα διαμάντι μέσα σε κάποιο από τα κουτιά. Το βραβείο τύπου v είναι το πιο φθηνό από όλα: ένα γλειφιτζούρι. Για να γίνει το παιχνίδι πιο συναρπαστικό, το πλήθος των φθηνών βραβείων είναι πολύ μεγαλύτερο από το πλήθος των ακριβών. Πιο συγκεκριμένα, για κάθε t τέτοιο ώστε $2 \leq t \leq v$ γνωρίζουμε το εξής: αν υπάρχουν k βραβεία τύπου $t-1$, τότε υπάρχουν *αυστηρά περισσότερα* από k^2 βραβεία τύπου t.

Ο σκοπός σου είναι να βρείς το διαμάντι. Στο τέλος του παιχνιδιού θα πρέπει να ανοίξεις ένα κουτί και θα πάρεις το βραβείο που αυτό περιέχει. Πριν όμως χρειαστεί να διαλέξεις ποιο κουτί θα ανοίξεις, μπορείς να ρωτήσεις τον Rambod, τον οικοδεσπότη του παιχνιδιού, μερικές ερωτήσεις. Για κάθε ερώτηση, επιλέγεις κάποιο κουτί i. Η απάντηση που θα σου δώσει ο Rambod είναι ένας πίνακας a αποτελούμενος από δύο ακέραιους. Η σημασία τους είναι η ακόλουθη:

- Μεταξύ των κουτιών που βρίσκονται στα αριστερά του κουτιού i υπάρχουν ακριβώς $a[0]$ κουτιά που περιέχουν ακριβότερα βραβεία από αυτό στο κουτί i.
- Μεταξύ των κουτιών που βρίσκονται στα δεξιά του κουτιού i υπάρχουν ακριβώς $a[1]$ κουτιά που περιέχουν ακριβότερα βραβεία από αυτό στο κουτί i.

Για παράδειγμα, έστω ότι $n = 8$ και ότι για την ερώτησή σου, διαλέγεις το κουτί $i = 2$. Έστω επίσης ότι ο Rambod απαντάει ότι $a = [1, 2]$. Η σημασία αυτής της απάντησης είναι:

- Ακριβώς ένα από τα κουτιά 0 και 1 περιέχει βραβείο ακριβότερο από αυτό στο κουτί 2.
- Ακριβώς δύο από τα κουτιά $3, 4, \ldots, 7$ περιέχουν βραβεία ακριβότερα από αυτό στο κουτί 2.

Η δουλειά σου είναι να βρεις το κουτί που περιέχει το διαμάντι ρωτώντας μόνο λίγες ερωτήσεις.

Λεπτομέρειες υλοποίησης

Πρέπει να υλοποιήσεις την εξής διαδικασία:

```
int find_best(int n)
```

- Αυτή η διαδικασία καλείται ακριβώς μία φορά από τον βαθμολογητή

- n: το πλήθος των κουτιών
- Η διαδικασία πρέπει να επιστρέφει τον αριθμό του κουτιού που περιέχει το διαμάντι, δηλαδή, έναν μοναδικό ακέραιο d ($0 \leq d \leq n-1$) τέτοιο ώστε το κουτί d να περιέχει βραβείο τύπου 1.

Η παραπάνω διαδικασία μπορεί να κάνει κλήσεις στην ακόλουθη:

```
int[] ask(int i)
```

- i: ο αριθμός του κουτιού για το οποίο επιλέγεις να ρωτήσεις. Η τιμή του i πρέπει να είναι μεταξύ 0 και $n-1$.
- Η διαδικασία επιστρέφει έναν πίνακα a με 2 στοιχεία. Εδώ το $a[0]$ είναι το πλήθος των ακριβότερων βραβείων που βρίσκονται στα αριστερά του κουτιού i, και $a[1]$ είναι το πλήθος των ακριβότερων βραβείων που βρίσκονται στα δεξιά του κουτιού i.

Παράδειγμα

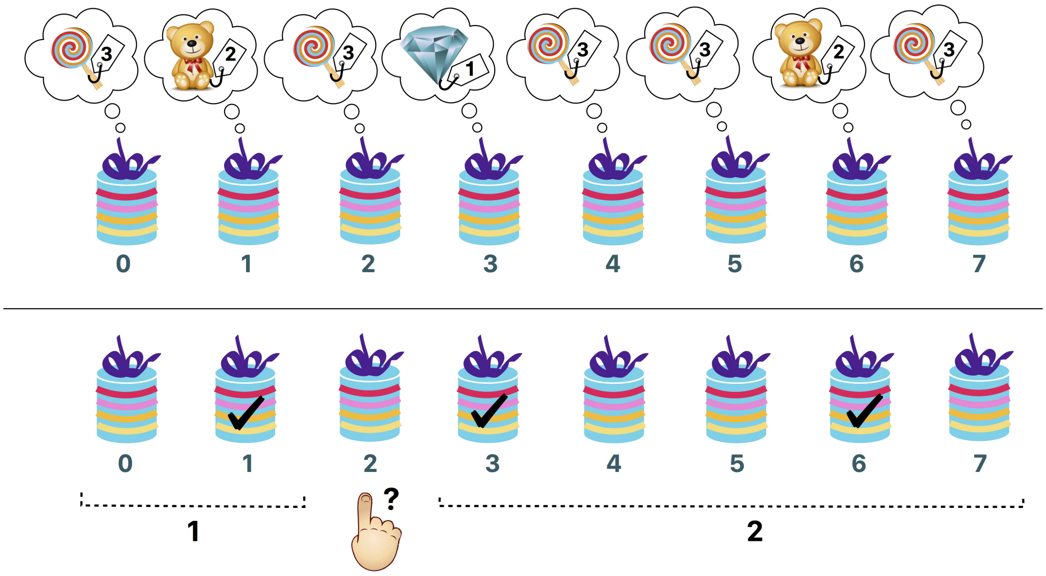
Ο βαθμολογητής κάνει την παρακάτω κλήση:

```
find_best(8)
```

Υπάρχουν $n=8$ κουτιά. Ας υποθέσουμε ότι οι τύποι των βραβείων είναι $[3,2,3,1,3,3,2,3]$. Όλες οι δυνατές κλήσεις στης διαδικασία `ask` και τα αντίστοιχα αποτελέσματα δίνονται παρακάτω.

- `ask(0)` επιστρέφει $[0,3]$
- `ask(1)` επιστρέφει $[0,1]$
- `ask(2)` επιστρέφει $[1,2]$
- `ask(3)` επιστρέφει $[0,0]$
- `ask(4)` επιστρέφει $[2,1]$
- `ask(5)` επιστρέφει $[2,1]$
- `ask(6)` επιστρέφει $[1,0]$
- `ask(7)` επιστρέφει $[3,0]$

Σε αυτό το παράδειγμα, το διαμάντι βρίσκεται στο κουτί 3. Η διαδικασία `find_best` επομένως πρέπει να επιστρέψει 3.



Το παράδειγμα φαίνεται στο παραπάνω σχήμα. Το πάνω μέρος του σχήματος δείχνει τους τύπους των βραβείων κάθε κουτιού. Το κάτω μέρος δείχνει την ερώτηση `ask(2)`. Τα σημειωμένα κουτιά είναι αυτά που περιέχουν ακριβότερα βραβεία από αυτό του κουτιού 2.

Περιορισμοί

- $3 \leq n \leq 200\,000$.
- Ο τύπος του βραβείου σε κάθε κουτί είναι μεταξύ 1 και v, συμπεριλαμβανομένων.
- Υπάρχει ακριβώς ένα βραβείο τύπου 1.
- Για κάθε $2 \leq t \leq v$, αν υπάρχουν k βραβεία τύπου $t-1$, τότε υπάρχουν *αυστηρά περισσότερα* από k^2 βραβεία τύπου t.

Υποπροβλήματα και βαθμολογία

Σε ορισμένες περιπτώσεις ελέγχου, η συμπεριφορά του βαθμολογητή είναι προσαρμοστική. Αυτό σημαίνει ότι σε αυτές τις περιπτώσεις ελέγχου ο βαθμολογητής δεν θα έχει προαποφασισμένη τη σειρά των βραβείων. Αντίθετα, οι απαντήσεις που θα δίνει ενδέχεται να εξαρτώνται από τις ερωτήσεις που του γίνονται. Είναι εγγυημένο ότι οι απαντήσεις του βαθμολογητή θα είναι τέτοιες ώστε μετά από κάθε μία θα υπάρχει τουλάχιστον μία ακολουθία βραβείων που να είναι συνεπής με όλες τις απαντήσεις που έχουν δοθεί μέχρι τότε.

1. (20 βαθμοί) Υπάρχει ακριβώς 1 διαμάντι και $n-1$ γλειφιτζούρια (επομένως, $v=2$). Μπορείς να καλέσεις τη διαδικασία `ask` το πολύ $10\,000$ φορές.
2. (80 βαθμοι) Κανένας πρόσθετος περιορισμός.

Στο υποπρόβλημα 2 μπορείς να πάρεις μερική βαθμολογία. Αν q είναι το μέγιστο πλήθος κλήσεων προς τη διαδικασία `ask` που γίνεται μεταξύ όλων των περιπτώσεων ελέγχου αυτού του υποπροβλήματος, τότε ο βαθμός σου για αυτό το υποπρόβλημα υπολογίζεται σύμφωνα με τον παρακάτω πίνακα:

Ερωτήσεις	Βαθμός
$10\,000 < q$	0 (στο CMS αναφέρεται 'Wrong Answer')
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

Υπόδειγμα βαθμολογητή

Ο βαθμολογητής που σας δίνεται ως υπόδειγμα δεν είναι προσαρμοστικός. Αντίθετα, διαβάζει και χρησιμοποιεί έναν προκαθορισμένο πίνακα p με τους τύπους των βραβείων. Για κάθε $0 \leq b \leq n - 1$, ο τύπος βραβείου στο κουτί b δίνεται από το $p[b]$.

Το υπόδειγμα βαθμολογητή διαβάζει την είσοδο με την παρακάτω μορφή:

- γραμμή 1: n
- γραμμή 2: $p[0]\ \ p[1]\ \ \ldots\ p[n-1]$

Το υπόδειγμα βαθμολογητή τυπώνει μία γραμμή που περιέχει την τιμή επιστροφής της διαδικασίας `find_best` και το πλήθος των κλήσεων προς τη διαδικασία `ask`.