



Χιλιάδες Νησιά

Τα "Χιλιάδες Νησιά" είναι ένα σύμπλεγμα όμορφων νησιών στη Θάλασσα της Ιάβας. Το σύμπλεγμα αποτελείται από N νησιά, αριθμημένα από 0 έως $N - 1$.

Υπάρχουν M κανό, αριθμημένα από 0 έως $M - 1$, τα οποία μπορούν να χρησιμοποιηθούν για πλεύση μεταξύ των νησιών. Για κάθε i τέτοιο ώστε $0 \leq i \leq M - 1$, το κανό i μπορεί να αγκυροβολήσει είτε στο νησί $U[i]$ είτε στο $V[i]$, και μπορεί να χρησιμοποιηθεί για να πλεύσει μεταξύ των νησιών $U[i]$ και $V[i]$. Συγκεκριμένα, όταν το κανό είναι αγκυροβολημένο στο νησί $U[i]$, μπορεί να χρησιμοποιηθεί για να πλεύσει από το νησί $U[i]$ στο νησί $V[i]$, και μετά το κανό αγκυροβολεί στο νησί $V[i]$. Παρόμοια, όταν το κανό είναι αγκυροβολημένο στο νησί $V[i]$, μπορεί να χρησιμοποιηθεί για να πλεύσει από το νησί $V[i]$ στο νησί $U[i]$, και μετά το κανό αγκυροβολεί στο νησί $U[i]$. Αρχικά, το κανό είναι αγκυροβολημένο στο νησί $U[i]$. Είναι δυνατό πολλά κανό να μπορούν να χρησιμοποιηθούν για να πλεύσουν μεταξύ του ίδιου ζεύγους νησιών. Επίσης είναι δυνατό πολλά κανό να έχουν αγκυροβολήσει στο ίδιο νησί.

Για λόγους ασφάλειας, ένα κανό χρειάζεται να συντηρηθεί μετά από κάθε πλεύση του, γεγονός που απαγορεύει το ίδιο κανό να πλεύσει δύο φορές στη σειρά. Δηλαδή μετά τη χρήση κάποιου κανό i , ένα άλλο κανό πρέπει να χρησιμοποιηθεί πριν το κανό i να μπορέσει να ξαναχρησιμοποιηθεί.

Η Bu Dengklek επιθυμεί να σχεδιάσει μια διαδρομή για ένα ταξίδι περνώντας από κάποια από τα νησιά. Η διαδρομή του ταξιδιού της είναι **έγκυρη** εάν και μόνο αν ικανοποιούνται οι παρακάτω συνθήκες.

- Η Bu Dengklek ξεκινά και τελειώνει το ταξίδι της στο νησί 0.
- Η Bu Dengklek επισκέπτεται τουλάχιστον ένα νησί διαφορετικό από το νησί 0.
- Μετά το τέλος της διαδρομής, κάθε κανό είναι αγκυροβολημένο στο ίδιο νησί, όπου ήταν πριν ξεκινήσει το ταξίδι. Δηλ. το κανό i , για κάθε i τέτοιο ώστε $0 \leq i \leq M - 1$, πρέπει να είναι αγκυροβολημένο στο νησί $U[i]$.

Βοηθήστε την Bu Dengklek να βρει μια οποιαδήποτε έγκυρη διαδρομή για το ταξίδι της, η οποία θα περιλαμβάνει το πολύ 2 000 000 πλεύσεις, ή προσδιορίστε ότι δεν υπάρχει τέτοια έγκυρη διαδρομή. Μπορεί να αποδειχθεί ότι σύμφωνα με τους περιορισμούς που δίνονται σε αυτό το πρόβλημα (βλ. ενότητα Περιορισμοί), εάν υπάρχει μια έγκυρη διαδρομή, υπάρχει επίσης και μια έγκυρη διαδρομή, η οποία δεν περιλαμβάνει περισσότερες από 2 000 000 πλεύσεις.

Λεπτομέρειες Υλοποίησης

Πρέπει να υλοποιήσετε την παρακάτω συνάρτηση:

```
union(bool, int[]) find_journey(int N, int M, int[] U, int[] V)
```

- N : το πλήθος των νησιών.
- M : το πλήθος των κανό.
- U, V : πίνακες μήκους M που περιγράφουν τα κανό.
- Αυτή η συνάρτηση πρέπει να επιστρέφει είτε μια λογική τιμή (boolean) ή έναν πίνακα από ακέραιους.
- Εάν δεν υπάρχει έγκυρη διαδρομή, η συνάρτηση πρέπει να επιστρέψει false.
- Εάν υπάρχει έγκυρη διαδρομή έχετε δύο επιλογές:
- Για να λάβετε την πλήρη βαθμολογία, η συνάρτηση πρέπει να επιστρέψει έναν πίνακα με το πολύ 2 000 000 ακραίους που αντιπροσωπεύουν μια έγκυρη διαδρομή. Πιο συγκεκριμένα, τα στοιχεία αυτού του πίνακα πρέπει να είναι οι αριθμοί των κανό, τα οποία χρησιμοποιήθηκαν στη διαδρομή (με τη σειρά που αυτά χρησιμοποιήθηκαν).
- Για να λάβετε μερική βαθμολογία η συνάρτηση πρέπει να επιστρέψει true, έναν πίνακα με περισσότερους από 2 000 000 ακραίους ή έναν πίνακα ακεραίων που δεν περιγράφουν μια έγκυρη διαδρομή (δείτε την ενότητα Υποπροβλήματα για περισσότερες λεπτομέρειες).
- Αυτή η συνάρτηση καλείται ακριβώς μία φορά.

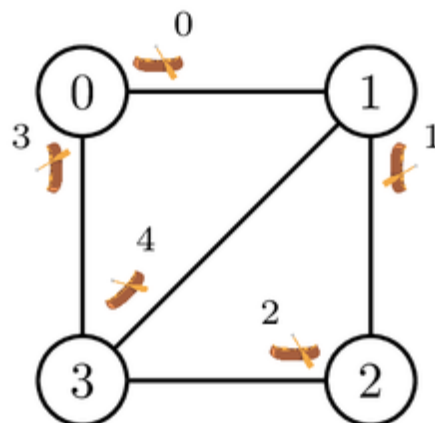
Παραδείγματα

Παράδειγμα 1

Θεωρείστε την ακόλουθη κλήση:

```
find_journey(4, 5, [0, 1, 2, 0, 3], [1, 2, 3, 3, 1])
```

Τα νησιά και τα κανό απεικονίζονται στην παρακάτω εικόνα.



Μια πιθανή έγκυρη διαδρομή είναι η ακόλουθη. Η Bu Dengklek αρχικά πλέει με τα κανό 0, 1, 2, και 4 σε αυτή τη σειρά. Ως αποτέλεσμα, βρίσκεται στο νησί 1. Μετά από αυτό, η Bu Dengklek μπορεί να πλεύσει με το κανό 0 ξανά, καθώς αυτό είναι προσωρινά αγκυροβολημένο στο νησί 1 και το τελευταίο κανό που χρησιμοποίησε δεν είναι το κανό 0. Αφού πλεύσει με το κανό 0 ξανά, η Bu Dengklek βρίσκεται τώρα στο νησί 0. Ωστόσο, τα κανό 1, 2 και 4 δεν είναι αγκυροβολημένα στα ίδια νησιά όπου ήταν πριν από το ταξίδι. Η Bu Dengklek έπειτα συνεχίζει το ταξίδι της πλέοντας με τα κανό 3, 2, 1, 4, και 3 ξανά. Η Bu Dengklek βρίσκεται πίσω στο νησί 0 και όλα τα κανό είναι αγκυροβολημένα στα ίδια νησιά όπως πριν από το ταξίδι.

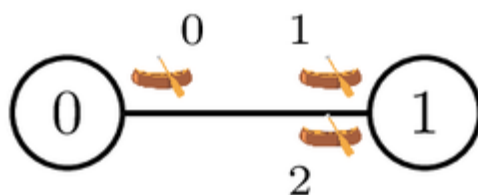
Επομένως, η επιστρεφόμενη τιμή $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$ αναπαριστά μια έγκυρη διαδρομή.

Παράδειγμα 2

Θεωρείστε την ακόλουθη κλήση:

```
find_journey(2, 3, [0, 1, 1], [1, 0, 0])
```

Τα νησιά και τα κανό απεικονίζονται στην παρακάτω εικόνα.



Η Bu Dengklek μπορεί μόνο να ξεκινήσει με το να πλεύσει με το κανό 0, κι έπειτα μπορεί να πλεύσει είτε με το κανό 1 είτε με το κανό 2. Σημειώστε ότι δεν μπορεί να πλεύσει με το κανό 0 δύο φορές στη σειρά. Και στις δύο περιπτώσεις, η Bu Dengklek βρίσκεται πίσω στο νησί 0. Ωστόσο, τα κανό δεν είναι αγκυροβολημένα στα ίδια νησιά όπου ήταν πριν από το ταξίδι και η Bu Dengklek δεν μπορεί στη συνέχεια να πλεύσει με κανένα κανό, μιας και το μόνο αγκυροβολημένο κανό στο νησί 0 είναι αυτό που μόλις χρησιμοποίησε. Καθώς δεν υπάρχει έγκυρη διαδρομή, η συνάρτηση πρέπει να επιστρέψει `false`.

Περιορισμοί

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$ και $0 \leq V[i] \leq N - 1$ (για κάθε i τέτοιο ώστε $0 \leq i \leq M - 1$)
- $U[i] \neq V[i]$ (για κάθε i τέτοιο ώστε $0 \leq i \leq M - 1$)

Υποπροβλήματα

1. (5 βαθμοί) $N = 2$
2. (5 βαθμοί) $N \leq 400$. Για κάθε ζευγάρι διακριτών (διαφορετικών) νησιών x και y ($0 \leq x < y \leq N - 1$), υπάρχουν ακριβώς δύο κανό που μπορούν να χρησιμοποιηθούν για να πλεύσουν μεταξύ αυτών. Ένα από αυτά είναι αγκυροβολημένο στο νησί x , και το άλλο είναι αγκυροβολημένο στο νησί y .
3. (21 βαθμοί) $N \leq 1000$, M είναι άρτιος, και για κάθε **άρτιο** i τέτοιο ώστε $0 \leq i \leq M - 1$, τα κανό i και $i + 1$ μπορούν και τα δύο να χρησιμοποιηθούν για να πλεύσουν μεταξύ των νησιών $U[i]$ και $V[i]$. Το κανό i είναι αρχικά αγκυροβολημένο στο νησί $U[i]$ και το κανό $i + 1$ είναι αρχικά αγκυροβολημένο στο νησί $V[i]$. Τυπικά, $U[i] = V[i + 1]$ και $V[i] = U[i + 1]$.
4. (24 βαθμοί) $N \leq 1000$, M είναι άρτιος, και για κάθε **άρτιο** i τέτοιο ώστε $0 \leq i \leq M - 1$, τα κανό i και $i + 1$ μπορούν και τα δύο να χρησιμοποιηθούν για να πλεύσουν μεταξύ των νησιών $U[i]$ και $V[i]$. Και τα δυο κανό είναι αρχικά αγκυροβολημένα στο νησί $U[i]$. Τυπικά, $U[i] = U[i + 1]$ και $V[i] = V[i + 1]$.
5. (45 βαθμοί) Κανένας επιπλέον περιορισμός.

Για κάθε αρχείο ελέγχου (test case), στο οποίο υπάρχει μια έγκυρη διαδρομή, η λύση σας:

- λαμβάνει πλήρη βαθμολογία εάν επιστρέψει μια έγκυρη διαδρομή,
- λαμβάνει 35% της βαθμολογίας εάν επιστρέψει true, έναν πίνακα με περισσότερους από 2 000 000 ακεραίους, ή έναν πίνακα που δεν περιγράφει μια έγκυρη διαδρομή,
- λαμβάνει 0 βαθμούς σε κάθε άλλη περίπτωση.

Για κάθε αρχείο ελέγχου (test case), στο οποίο δεν υπάρχει μια έγκυρη διαδρομή, η λύση σας:

- λαμβάνει πλήρη βαθμολογία εάν επιστρέψει false,
- λαμβάνει 0 βαθμούς σε κάθε άλλη περίπτωση.

Σημειώστε ότι η τελική βαθμολογία για κάθε υποπρόβλημα (subtask) είναι το ελάχιστο των βαθμών για τα αρχεία ελέγχου (test case) του υποπροβλήματος.

Υπόδειγμα Βαθμολογητή

Ο βαθμολογητής διαβάσει την είσοδο με την ακόλουθη μορφή:

- γραμμή 1 : $N M$
- γραμμή $2 + i$ ($0 \leq i \leq M - 1$) : $U[i] V[i]$

Ο βαθμολογητής εκτυπώνει τις απαντήσεις σας με την ακόλουθη μορφή:

- Εάν η `find_journey` επιστρέψει `bool`:
 - γραμμή 1 : 0
 - γραμμή 2 : 0 εάν η `find_journey` επιστρέψει `false`, ή 1 σε διαφορετική περίπτωση.
- Εάν η `find_journey` επιστρέψει `int[]`, συμβολίζουμε τα στοιχεία του πίνακα αυτού με $c[0], c[1], \dots, c[k - 1]$. Ο βαθμολογητής εκτυπώνει:

- γραμμή 1 : 1
- γραμμή 2 : k
- γραμμή 3 : $c[0] \ c[1] \ \dots \ c[k - 1]$