



Werewolf

Στην επαρχία Ιμπαράκι της Ιαπωνίας υπάρχουν N πόλεις και M δρόμοι. Οι πόλεις είναι αριθμημένες από το 0 μέχρι το $N - 1$, σε αύξουσα σειρά ανάλογα με τον πληθυσμό τους. Κάθε δρόμος συνδέει ένα ζεύγος διαφορετικών μεταξύ τους πόλεων και είναι διπλής κατεύθυνσης. Μπορείτε να ταξιδέψετε από κάθε πόλη σε οποιαδήποτε άλλη πόλη χρησιμοποιώντας έναν ή περισσότερους από αυτούς τους δρόμους.

Έχετε προγραμματίσει Q ταξίδια, αριθμημένα από το 0 μέχρι το $Q - 1$. Το ταξίδι i ($0 \leq i \leq Q - 1$) έχει ως αφετηρία την πόλη S_i και ως τερματισμό την πόλη E_i .

Είστε ένας λυκάνθρωπος. Έχετε δυο μορφές: **ανθρώπινη μορφή** και **μορφή λύκου**. Στη αρχή κάθε ταξιδιού είστε στην ανθρώπινη μορφή. Στο τέλος κάθε ταξιδιού θα πρέπει να είστε στη μορφή λύκου. Κατά την διάρκεια του ταξιδιού θα πρέπει να **μεταμορφωθείτε** (να αλλάξετε από την ανθρώπινη μορφή σε μορφή λύκου) ακριβώς μία φορά και αυτό θα πρέπει να γίνει καθώς βρίσκεστε σε κάποια πόλη (πιθανώς την S_i , την E_i ή κάποια ενδιάμεση στο ταξίδι σας).

Η ζωή του λυκάνθρωπου είναι δύσκολη. Πρέπει να αποφεύγετε τις αραιοκατοικημένες πόλεις, όταν είστε στην ανθρώπινη μορφή, και τις πυκνοκατοικημένες πόλεις, όταν είστε στη μορφή λύκου. Για κάθε ταξίδι i υπάρχουν δύο ακέραιοι αριθμοί L_i και R_i ($0 \leq L_i \leq R_i \leq N - 1$) που σας δείχνουν ποιες πόλεις πρέπει να αποφύγετε. Συγκεκριμένα, πρέπει να αποφύγετε τις πόλεις $0, 1, \dots, L_i - 1$ όταν είστε στην ανθρώπινη μορφή και να αποφύγετε τις πόλεις $R_i + 1, R_i + 2, \dots, N - 1$ όταν είστε στη μορφή λύκου. Αυτό σημαίνει ότι στο ταξίδι i μπορείτε να μεταμορφωθείτε μόνο αν βρίσκεστε σε κάποια από τις πόλεις $L_i, L_i + 1, \dots, R_i$.

Ζητείται να βρείτε, για κάθε ταξίδι, αν είναι εφικτό να ταξιδέψετε από την πόλη S_i στη πόλη E_i με τρόπο που να ικανοποιεί τους παραπάνω περιορισμούς. Η διαδρομή που θα ακολουθήσετε μπορεί να είναι αυθαίρετα μεγάλη.

Λεπτομέρειες Υλοποίησης

Θα πρέπει να υλοποιήσετε την ακόλουθη συνάρτηση:

```
int[] check_validity(int N, int[] X, int[] Y, int[] S, int[] E, int[] L, int[] R)
```

- N : το πλήθος των πόλεων.

- X και Y : πίνακες μεγέθους M . Για κάθε j ($0 \leq j \leq M - 1$), η πόλη $X[j]$ είναι απευθείας συνδεδεμένη με την πόλη $Y[j]$ με έναν δρόμο.
- S , E , L , και R : πίνακες μεγέθους Q που αναπαριστούν τα ταξίδια.

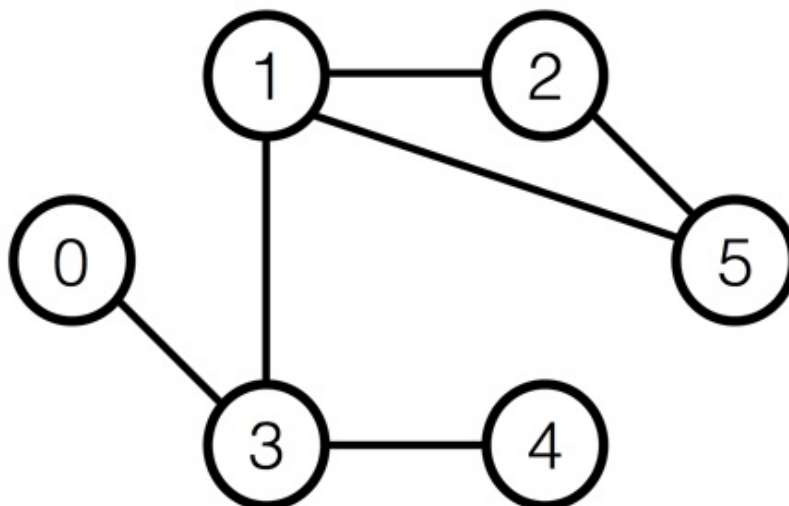
Προσέξτε ότι οι τιμές των M και Q είναι τα μεγέθη των πινάκων και μπορούν να βρεθούν όπως εξηγείται στο φυλλάδιο των Σημειώσεων Υλοποίησης.

Η συνάρτηση `check_validity` καλείται ακριβώς μια φορά για κάθε περίπτωση ελέγχου. Αυτή η συνάρτηση θα πρέπει να επιστρέφει έναν πίνακα ακεραίων A , μεγέθους Q . Η τιμή του A_i ($0 \leq i \leq Q - 1$) πρέπει να είναι 1 αν είναι εφικτό το ταξίδι i με τους παραπάνω περιορισμούς, διαφορετικά 0.

Παράδειγμα

Έστω $N = 6$, $M = 6$, $Q = 3$, $X = [5, 1, 1, 3, 3, 5]$, $Y = [1, 2, 3, 4, 0, 2]$, $S = [4, 4, 5]$, $E = [2, 2, 4]$, $L = [1, 2, 3]$, και $R = [2, 2, 4]$.

Ο βαθμολογητής καλεί τη `check_validity(6, [5, 1, 1, 3, 3, 5], [1, 2, 3, 4, 0, 2], [4, 4, 5], [2, 2, 4], [1, 2, 3], [2, 2, 4])`.



Για το ταξίδι 0, μπορείτε να ταξιδέψετε από την πόλη 4 στην πόλη 2 ως εξής:

- Ξεκινήστε στη πόλη 4 (ανθρώπινη μορφή)
- Μετακινηθείτε στην πόλη 3 (ανθρώπινη μορφή)
- Μετακινηθείτε στην πόλη 1 (ανθρώπινη μορφή)
- Μεταμορφωθείτε στη μορφή λύκου (μορφή λύκου)
- Μετακινηθείτε στη πόλη 2 (μορφή λύκου)

Για τα ταξίδια 1 και 2, δεν μπορείτε να ταξιδέψετε μεταξύ των πόλεων.

Έτσι, το πρόγραμμα επιστρέφει `[1, 0, 0]`.

Τα αρχεία `sample-01-in.txt` και `sample-01-out.txt` στο συμπιεσμένο πακέτο αντιστοιχούν σε αυτό το παράδειγμα. Περιέχονται και άλλα παραδείγματα εισόδου/

εξόδου στο πακέτο.

Περιορισμοί

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Για κάθε $0 \leq j \leq M - 1$:
 - $0 \leq X_j \leq N - 1$
 - $0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- Μπορείτε να ταξιδέψετε από οποιαδήποτε πόλη σε οποιαδήποτε άλλη χρησιμοποιώντας δρόμους.
- Κάθε ζεύγος πόλεων συνδέεται απευθείας το πολύ με ένα δρόμο. Με άλλα λόγια, για κάθε $0 \leq j < k \leq M - 1$, $(X_j, Y_j) \neq (X_k, Y_k)$ και $(Y_j, X_j) \neq (X_k, Y_k)$.
- Για κάθε $0 \leq i \leq Q - 1$
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$
 - $L_i \leq R_i$

Υποπροβλήματα

1. (7 βαθμοί) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 βαθμοί) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 βαθμοί) $M = N - 1$ και καμία πόλη δεν συνδέεται απευθείας με περισσότερες από 2 άλλες πόλεις (οι πόλεις είναι ενωμένες σε μια γραμμή)
4. (51 βαθμοί) Χωρίς επιπρόσθετους περιορισμούς

Υποδειγματικός βαθμολογητής

Ο υποδειγματικός βαθμολογητής διαβάζει την είσοδο στην ακόλουθη μορφή:

- γραμμή 1: $N M Q$
- γραμμή $2 + j$ ($0 \leq j \leq M - 1$): $X_j Y_j$
- γραμμή $2 + M + i$ ($0 \leq i \leq Q - 1$): $S_i E_i L_i R_i$

Ο υποδειγματικός βαθμολογητής τυπώνει το αποτέλεσμα της `check_validity` στην ακόλουθη μορφή:

- γραμμή $1 + i$ ($0 \leq i \leq Q - 1$): A_i