



Παιχνίδι

Ο Jian-Jia είναι ένα νεαρό αγόρι που αγαπά να παίζει παιχνίδια. Όταν του κάνουν μια ερώτηση, προτιμά να παίζει παιχνίδια παρά να απαντά απευθείας. Ο Jian-Jia συνάντησε τη φίλη του Mei-Yu και της είπε για το δίκτυο πτήσεων της Ταϊβαν. Υπάρχουν n πόλεις στη Ταϊβαν (αριθμημένες $0, \dots, n - 1$), εκ των οποίων μερικές συνδέονται μεταξύ τους με πτήσεις. Κάθε πτήση συνδέει δύο πόλεις και μπορεί να εκτελεστεί και προς τις δύο κατευθύνσεις.

Η Mei-Yu ρώτησε τον Jian-Jia αν είναι δυνατό να πάει αεροπορικά μεταξύ δύο πόλεων (είτε απευθείας ή έμμεσα). Ο Jian-Jia δεν ήθελε να αποκαλύψει την απάντηση, αντίθετα πρότεινε να παίξουν ένα παιχνίδι. Η Mei-Yu μπορεί να του κάνει ερωτήσεις της μορφής "Οι πόλεις x και y συνδέονται *άμεσα* με πτήση;" και ο Jian-Jia θα απαντά αυτές τις ερωτήσεις αμέσως. Η Mei-Yu θα ρωτήσει για κάθε ζεύγος πόλεων ακριβώς μια φορά, κάνοντας στο σύνολο $r = n(n - 1)/2$ ερωτήσεις. Η Mei-Yu κερδίζει το παιχνίδι αν, αφού λάβει τις απαντήσεις για τις πρώτες i ερωτήσεις, για κάποιο $i < r$, μπορεί να συμπεράνει κατα πόσο το δίκτυο είναι συνδεδεμένο, δηλαδή κατά πόσο είναι δυνατό να ταξιδέψει κανείς μεταξύ οποιωνδήποτε δύο πόλεων (είτε απευθείας ή έμμεσα). Αλλιώς, αν δηλαδή χρειαστεί όλες τις r ερωτήσεις, τότε ο νικητής είναι ο Jian-Jia.

Για να γίνει το παιχνίδι περισσότερο διασκεδαστικό για τον Jian-Jia, οι δύο φίλοι συμφώνησαν ότι ο Jian-Jia δικαιούται να αγνοήσει το πραγματικό δίκτυο πτήσεων της Ταϊβαν, και να φευρίσκει το δίκτυο όπως προχωράει το παιχνίδι, διαλέγοντας τις απαντήσεις του βάσει των προηγούμενων ερωτήσεων της Mei-Yu. Ο σκοπός σας είναι να βοηθήσετε τον Jian-Jia να κερδίσει το παιχνίδι, αποφασίζοντας πώς πρέπει να απαντήσει τις ερωτήσεις.

Παραδείγματα

Εξηγούμε τους κανόνες του παιχνιδιού με τρία παραδείγματα. Κάθε παράδειγμα έχει $n = 4$ πόλεις και $r = 6$ γύρους ερώτησης και απάντησης.

Στο πρώτο παράδειγμα (βλ. παρακάτω πίνακα), ο Jian-Jia *χάνει* γιατί μετά το γύρο 4, η Mei-Yu γνωρίζει σίγουρα ότι κάποιος μπορεί να ταξιδέψει αεροπορικά μεταξύ οποιωνδήποτε δύο πόλεων, άσχετα πώς θα απαντήσει ο Jian-Jia τις ερωτήσεις 5 ή 6.

γύρος	ερώτηση	απάντηση
1	0, 1	ναι
2	3, 0	όχι
3	1, 2	όχι
4	0, 2	ναι
-----	-----	-----
5	3, 1	όχι
6	2, 3	όχι

Στο επόμενο παράδειγμα, η Mei-Yu μπορεί να αποδείξει ότι μετά τον γύρο 3 ανεξάρτητα από το πώς θα απαντήσει ο Jian-Jia τις ερωτήσεις 4, 5, ή 6, δεν μπορεί κανείς να ταξιδέψει αεροπορικά μεταξύ των πόλεων 0 και 1, έτσι ο Jian-Jia χάνει και πάλι.

γύρος	ερώτηση	απάντηση
1	0, 3	όχι
2	2, 0	όχι
3	0, 1	όχι
-----	-----	-----
4	1, 2	ναι
5	1, 3	ναι
6	2, 3	ναι

Στο τελικό παράδειγμα, η Mei-Yu δεν μπορεί να αποφασίσει κατα πόσο κάποιος μπορεί να ταξιδέψει αεροπορικά μεταξύ οποιωνδήποτε δύο πόλεων μέχρι να απαντηθούν και οι έξι ερωτήσεις, έτσι ο Jian-Jia κερδίζει το παιχνίδι. Συγκεκριμένα, επειδή ο Jian-Jia απάντησε ναι στην τελευταία ερώτηση (στον πίνακα που ακολουθεί), είναι δυνατό να ταξιδέψει κανείς μεταξύ οποιωνδήποτε δύο πόλεων. Όμως, αν ο Jian-Jia απαντούσε όχι στην τελευταία ερώτηση, τότε θα ήταν αδύνατο.

γύρος	ερώτηση	απάντηση
1	0, 3	όχι
2	1, 0	ναι
3	0, 2	όχι
4	3, 1	ναι
5	1, 2	όχι
6	2, 3	ναι

Πρόβλημα

Γράψτε ένα πρόγραμμα που να βοηθά τον Jian-Jia να κερδίσει το παιχνίδι. Σημειώστε ότι ούτε η Mei-Yu αλλά ούτε ο Jian-Jia γνωρίζουν την στρατηγική του άλλου παίκτη. Η Mei-Yu μπορεί να κάνει ερωτήσεις για ζεύγη πόλεων με οποιαδήποτε σειρά, και ο Jian-Jia πρέπει να τις απαντάει άμεσα χωρίς να γνωρίζει τις ερωτήσεις που θα ακολουθήσουν. Πρέπει να υλοποιήσετε τις ακόλουθες δύο συναρτήσεις.

- `initialize(n)` -- Η συνάρτηση `initialize` θα κληθεί πρώτα από το σύστημα. Η παράμετρος n είναι το πλήθος των πόλεων.
- `hasEdge(u, v)` -- Στη συνέχεια θα κληθεί η συνάρτηση `hasEdge` $r = n(n - 1)/2$ φορές. Αυτές οι κλήσεις αντιπροσωπεύουν τις ερωτήσεις της Mei-Yu, με τη σειρά που τις κάνει. Πρέπει να απαντήσετε κατα πόσο υπάρχει απευθείας πτήση ανάμεσα στις πόλεις u και v . Συγκεκριμένα, η τιμή επιστροφής θα πρέπει να είναι 1 εαν υπάρχει απευθείας πτήση, διαφορετικά 0.

Υποπροβλήματα

Κάθε υποπρόβλημα αποτελείται από αρκετά παιχνίδια. Θα πάρετε μονάδες για ένα υποπρόβλημα μόνο αν το πρόγραμμά σας κερδίσει όλα τα παιχνίδια για τον Jian-Jia.

υποπρόβλημα	βαθμοί	n
1	15	$n = 4$
2	27	$4 \leq n \leq 80$
3	58	$4 \leq n \leq 1500$

Λεπτομέρειες υλοποίησης

Πρέπει να υποβάλετε ακριβώς ένα αρχείο, με όνομα `game.c`, `game.cpp` ή `game.pas`. Αυτό το αρχείο πρέπει να υλοποιεί τα υποπρόγραμματα όπως περιγράφονται παραπάνω, χρησιμοποιώντας τις παρακάτω επικεφαλίδες.

Πρόγραμμα C/C++

```
void initialize(int n);  
int hasEdge(int u, int v);
```

Πρόγραμμα Pascal

```
procedure initialize(n: longint);  
function hasEdge(u, v: longint): longint;
```

Ενδεικτικός βαθμολογητής

Ο ενδεικτικός βαθμολογητής διαβάζει την είσοδο με την ακόλουθη μορφή:

- Γραμμή 1: n
- οι επόμενες r γραμμές: κάθε γραμμή περιέχει δύο ακέραιους u και v που περιγράφουν μια ερώτηση για τις πόλεις u και v .