



Φίλοι

Κτιζουμε ένα κοινωνικό δίκτυο από n άτομα αριθμημένα $0, \dots, n - 1$. Κάποια ζεύγη ατόμων στο δίκτυο θα είναι φίλοι. Αν το άτομο x γίνει φίλος με το άτομο y , τότε το άτομο y θα γίνει επίσης φίλος με το άτομο x .

Τα άτομα προστίθενται στο δίκτυο σε n στάδια, τα οποία είναι επίσης αριθμημένα από 0 μέχρι $n - 1$. Το άτομο i προστίθεται στο στάδιο i . Στο στάδιο 0 , το άτομο 0 προστίθεται ως το μοναδικό άτομο του δικτύου. Στο κάθε ένα από τα επόμενα $n - 1$ στάδια, ένα άτομο προστίθεται στο δίκτυο από ένα οικοδεσπότη, ο οποίος μπορεί να είναι οποιοδήποτε άτομο που είναι ήδη στο δίκτυο. Στο στάδιο i ($0 < i < n$), ο οικοδεσπότης για αυτό το στάδιο μπορεί να προσθέσει το εισερχόμενο άτομο i στο δίκτυο χρησιμοποιώντας ένα από τα ακόλουθα τρία πρωτόκολλα:

- *ΕίμαιΔικόςΣουΦίλος* κάνει το άτομο i φίλο του οικοδεσπότη μόνο.
- *ΟιΦίλοιΜουΕίναιΔικοίΣουΦίλοι* κάνει το άτομο i φίλο με κάθε άτομο που είναι φίλος του οικοδεσπότη αυτή τη στιγμή. Σημειώστε ότι αυτό το πρωτόκολλο δεν κάνει το άτομο i φίλο του οικοδεσπότη.
- *ΕίμαστεΔικοίΣουΦίλοι* κάνει το άτομο i φίλο με τον οικοδεσπότη, και επίσης φίλο με κάθε άτομο που είναι φίλος του οικοδεσπότη αυτή τη στιγμή.

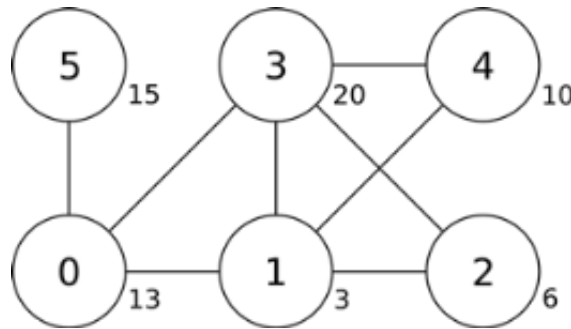
Όταν χτίσουμε το δίκτυο θα θέλαμε να διαλέξουμε ένα δείγμα για μια μελέτη, δηλαδή, να διαλέξουμε μια ομάδα ατόμων από το δίκτυο. Αφού οι φίλοι έχουν συνήθως όμοια ενδιαφέροντα, το δείγμα δεν θα πρέπει να περιλαμβάνει οποιοδήποτε ζεύγος ατόμων που να είναι φίλοι αναμεταξύ τους. Κάθε άτομο έχει ένα βαθμό εμπιστοσύνης μελέτης, ο οποίος εκφράζεται ως ένας θετικός ακαίρεος, και θα θέλαμε να βρούμε ένα δείγμα με μέγιστο συνολικό βαθμό εμπιστοσύνης.

Example

στάδιο	οικοδεσπότης	πρωτόκολλο	σχέσεις φίλων που προστίθενται
1	0	ΕίμαιΔικόςΣουΦίλος	(1, 0)
2	0	ΟιΦίλοιΜουΕίναιΔικοίΣουΦίλοι	(2, 1)
3	1	ΕίμαστεΔικοίΣουΦίλοι	(3, 1), (3, 0), (3, 2)
4	2	ΟιΦίλοιΜουΕίναιΔικοίΣουΦίλοι	(4, 1), (4, 3)
5	0	ΕίμαιΔικόςΣουΦίλος	(5, 0)

Αρχικά το δίκτυο περιέχει μόνο το άτομο 0 . Ο οικοδεσπότης του σταδίου 1 (άτομο 0) προσκαλεί το νέο άτομο 1 μέσω του πρωτοκόλλου *ΕίμαιΔικόςΣουΦίλος*, ως εκ τούτου γίνονται φίλοι. Ο οικοδεσπότης του σταδίου 2 (πάλι το άτομο 0) προσκαλεί το άτομο 2 με *ΟιΦίλοιΜουΕίναιΔικοίΣουΦίλοι*, το οποίο κάνει το άτομο 1 (τον μοναδικό φίλο του οικοδεσπότη) τον μοναδικό φίλο του ατόμου 2 . Ο οικοδεσπότης του σταδίου 3 (άτομο 1) προσθέτει το άτομο 3 μέσω *ΕίμαστεΔικοίΣουΦίλοι*, το οποίο κάνει το άτομο 3 φίλο του ατόμου 1 (ο οικοδεσπότης) και των

ατόμων 0 και 2 (φίλοι του οικοδεσπότη). Τα στάδιο 4 και 5 παρουσιάζονται επίσης στο πιο πάνω πίνακα. Το τελικό δίκτυο παρουσιάζεται στο ακόλουθο διάγραμμα, στο οποίο οι αριθμοί μέσα στους κύκλους δείχνουν τις ετικέτες των ατόμων, και οι αριθμοί δίπλα από τους κύκλους δείχνουν τον βαθμό εμπιστοσύνης μελέτης. Το δείγμα αποτελείται από τα άτομα 3 και 5 με συνολικό βαθμό εμπιστοσύνης $20 + 15 = 35$, το οποίο είναι το μέγιστο πιθανό σύνολο βαθμού εμπιστοσύνης.



Task

Δωσμένης της περιγραφής κάθε σταδίου και του βαθμού εμπιστοσύνης κάθε ατόμου, βρείτε ένα δείγμα με μέγιστο βαθμό εμπιστοσύνης. Εσείς πρέπει να υλοποιήσετε μόνο τη συντηση `findSample`.

- `findSample(n, confidence, host, protocol)`
 - `n`: το πλήθος ατόμων
 - `confidence`: πίνακας μεγέθους `n`; `confidence[i]` δίνει τον βαθμό εμπιστοσύνης του ατόμου `i`.
 - `host`: πίνακας μεγέθους `n`; `host[i]` δίνει τον οικοδεσπότη του σταδίου `i`.
 - `protocol`: πίνακας μεγέθους `n`; `protocol[i]` δίνει τον κωδικό πρωτοκόλλου που χρησιμοποιήθηκε στο στάδιο `i` ($0 < i < n$): 0 για ΕίμαιΔικόςΣουΦίλος, 1 για ΟιΦίλοιΜουΕίναιΔικοίΣουΦίλοι, και 2 για ΕίμαστεΔικοίΣουΦίλοι.
 - Αφού δεν υπάρχει οικοδεσπότης στο στάδιο 0, `host[0]` και `protocol[0]` δεν είναι καθορισμένα και δεν πρέπει να τα χρησιμοποιήσει το πρόγραμμα σας.
 - Η συνάρτηση πρέπει να επιστρέφει το μέγιστο δυνατό συνολικό βαθμό εμπιστοσύνης ενός δείγματος.

Subtasks

Κάποια υποπροβλήματα χρησιμοποιούν μόνο ένα υποσύνολο των πρωτοκόλλων, όπως δείχνει ο πιο κάτω πίνακας.

υποπρόβλημα	μονάδες	n	βαθμός εμπιστοσύνης	πρωτόκολλα που χρησιμοποιούν
1	11	$2 \leq n \leq 10$	$1 \leq \text{confidence} \leq 1,000,000$	Και τα τρία πρωτόκολλα
2	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Μόνο ΟιΦίλοιΜουΕίναιΔικοί
3	8	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Μόνο ΕίμαστεΔικοίΣου
4	19	$2 \leq n \leq 1,000$	$1 \leq \text{confidence} \leq 1,000,000$	Μόνο ΕίμαιΔικόςΣουΦ

υποπρόβλημα	μονάδες	n	βαθμός εμπιστοσύνης	πρωτόκολλα που χρησιμοποιούν
5	23	$2 \leq n \leq 1,000$	All confidence values are 1	Οι Φίλοι Μου Είναι Δικοί και Είμαι Δικός Σου Φίλος
6	31	$2 \leq n \leq 100,000$	$1 \leq \text{confidence} \leq 10,000$	Και τα τρία πρωτόκολλα

Implementation details

You have to submit exactly one file, called `friend.c`, `friend.cpp` or `friend.pas`. This file should implement the subprogram described above, using the following signatures. You also need to include a header file `friend.h` for C/C++ implementation.

C/C++ program

```
int findSample(int n, int confidence[], int host[], int protocol[]);
```

Pascal programs

```
function findSample(n: longint, confidence: array of longint, host: array of longint; protocol: array of longint): longint;
```

Sample grader

The sample grader reads the input in the following format:

- line 1: n
- line 2: $\text{confidence}[0], \dots, \text{confidence}[n-1]$
- line 3: $\text{host}[1], \text{protocol}[1], \text{host}[2], \text{protocol}[2], \dots, \text{host}[n-1], \text{protocol}[n-1]$

The sample grader will print the return value of `findSample`.