Σταθμοί (stations)

Η ραχοκοκαλιά του διαδικτύου της Σιγκαπούρης (SIB) αποτελείται από n σταθμούς, στους οποίους έχουν ανατεθεί **δείκτες** από το 0 μέχρι και το $n-1$. Υπάρχουν επίσης $n-1$ αμφίδρομοι σύνδεσμοι, αριθμημένοι από το 0 μέχρι και το $n-2$. Κάθε σύνδεσμος ενώνει δύο διαφορετικούς σταθμούς. Δύο σταθμοί που ενώνονται με ένα σύνδεσμο ονομάζονται γειτονικοί.

Μια διαδρομή από τον σταθμό x προς τον σταθμό y είναι μια ακολουθία διαφορετικών σταθμών $a_0, a_1, \cdots, a_p$, έτσι ώστε $a_0 = x$, $a_p = y$ και κάθε δύο διαδοχικοί σταθμοί στη διαδρομή είναι γειτονικοί. Υπάρχει **ακριβώς μία** διαδρομή από οποιονδήποτε σταθμό x σε οποιονδήποτε άλλο σταθμό y.

Οποιοσδήποτε σταθμός x μπορεί να στείλει ένα πακέτο δεδομένων σε οποιονδήποτε άλλο σταθμό y, που ονομάζεται **προορισμός** του πακέτου. Αυτό το πακέτο πρέπει να δρομολογηθεί κατά μήκος της μοναδικής διαδρομής από το x στο y ως εξής. Θεωρήστε έναν σταθμό z στον οποίο βρίσκεται ένα πακέτο με προορισμό το σταθμό y ($z \neq y$). Σε αυτή την περίπτωση, ο σταθμός z:

1. εκτελεί μια **διαδικασία δρομολόγησης** που καθορίζει το γειτονικό σταθμό του z ο οποίος βρίσκεται στη μοναδική διαδρομή από το z στο y, και
2. προωθεί το πακέτο σε αυτόν το γειτονικό σταθμό.

Ωστόσο, οι σταθμοί έχουν περιορισμένη μνήμη και δεν αποθηκεύουν ολόκληρη τη λίστα των συνδέσμων στο SIB για να την χρησιμοποιήσουν στη διαδικασία δρομολόγησης.

Η δική σας δουλειά είναι να υλοποιήσετε ένα σχήμα δρομολόγησης για το SIB, το οποίο να αποτελείται από δύο συναρτήσεις.

- Στην πρώτη συνάρτηση δίνονται ως εισόδοι το n, η λίστα των συνδέσμων στο SIB και ένας ακέραιος $k \geq n-1$. Αυτή η συνάρτηση αναθέτει σε κάθε σταθμό μια **μοναδική** ακέραια **ετικέτα** μεταξύ 0 και k, συμπεριλαμβανομένων.
- Η δεύτερη συνάρτηση υλοποιεί τη διαδικασία δρομολόγησης που θα ακολουθούν όλοι οι σταθμοί, αφού ανατεθούν οι ετικέτες. Της δίνονται **μόνο** τα ακόλουθα ως είσοδοι:
  - s, η **ετικέτα** του σταθμού όπου βρίσκεται ένα πακέτο,
  - t, η **ετικέτα** του σταθμού-προορισμού του πακέτου ($t \neq s$),
  - c, η λίστα των **ετικετών** όλων των γειτονικών σταθμών του s.

Η συνάρτηση πρέπει να επιστρέφει την **ετικέτα** του γεειτονικού σταθμού του s στον οποίο πρέπει να προωθηθεί το πακέτο.

Σε ένα υποπρόβλημα, η βαθμολογία της λύσης σας εξαρτάται από την τιμή της μέγιστης ετικέτας που έχει ανατεθεί σε οποιονδήποτε σταθμό (γενικά, πρέπει να προτιμώνται μικρές τιμές ετικετών).

Λεπτομέρειες υλοποίησης

Πρέπει να υλοποιήσετε τις παρακάτω συναρτήσεις:

```
int[] label(int n, int k, int[] u, int[] v)
```

- n: το πλήθος των σταθμών στο SIB.
- k: η μέγιστη ετικέτα που μπορεί να χρησιμοποιηθεί.
- u και v: δύο πίνακες μήκους $n-1$ που περιγράφουν τους συνδέσμους. Για κάθε i $(0 \leq i \leq n-2)$, ο σύνδεσμος i ενώνει τους σταθμους με δείκτες $u[i]$ και $v[i]$.
- Αυτή η συνάρτηση πρέπει να επιστρέφει έναν πίνακα L μήκους n. Για κάθε i $(0 \leq i \leq n-1)$ το $L[i]$ είναι η ετικέτα που ανατίθεται στον σταθμό με δείκτη i. Όλα τα στοιχεία του πίνακα L πρέπει να είναι μοναδικά και μεταξύ του 0 και του k, συμπεριλαμβανομένων.

```
int find_next_station(int s, int t, int[] c)
```

- s: η ετικέτα του σταθμού όπου βρίσκεται ένα πακέτο.
- t: η ετικέτα του σταθμού-προορισμού του πακέτου.
- c: ένας πίνακας που περιέχει τις ετικέτες όλων των γειτονικών σταθμών του s. Ο πίνακας c είναι ταξινομημένος σε αύξουσα σειρά.
- Αυτή η συνάρτηση πρέπει να επιστρέφει την ετικέτα του γειτονικού σταθμού του s στον οποίο πρέπει να προωθηθεί το πακέτο.

Κάθε περίπτωση ελέγχου (test case) περιλαμβάνει ένα ή περισσότερα ανεξάρτητα σενάρια (δηλαδή, διαφορετικές περιγραφές SIB). Για μία περίπτωση ελέγχου που περιλαμβάνει r σενάρια, ένα **πρόγραμμα** το οποίο καλεί τις πιο πάνω συναρτήσεις εκτελείται ακριβώς δύο φορές, ως ακολούθως.

Κατά την πρώτη εκτέλεση του προγράμματος:

- Η συνάρτηση `label` καλείται r φορές,
- οι επιστρεφόμενες ετικέτες αποθηκεύονται από το σύστημα αξιολόγησης, και
- η συνάρτηση `find_next_station` δεν καλείται.

Κατά τη δεύτερη εκτέλεση του προγράμματος:

- η συνάρτηση `find_next_station` μπορεί να κληθεί πολλές φορές. Σε κάθε κλήση ένα **τυχαίο** σενάριο επιλέγεται και οι ετικέτες που επιστρέφονται από την κλήση στη συνάρτηση `label` σε αυτό το σενάριο είναι αυτές που χρησιμοποιούνται σαν εισόδοι στη `find_next_station`.
- η συνάρτηση `label` δεν καλείται.

Συγκεκριμένα, οποιεσδήποτε πληροφορίες αποθηκεύονται σε στατικές ή καθολικές μεταβλητές κατά την πρώτη εκτέλεση του προγράμματος δεν είναι διαθέσιμες στη συνάρτηση `find_next_station`.

Παράδειγμα

Θεωρήστε την ακόλουθη συνάρτηση:

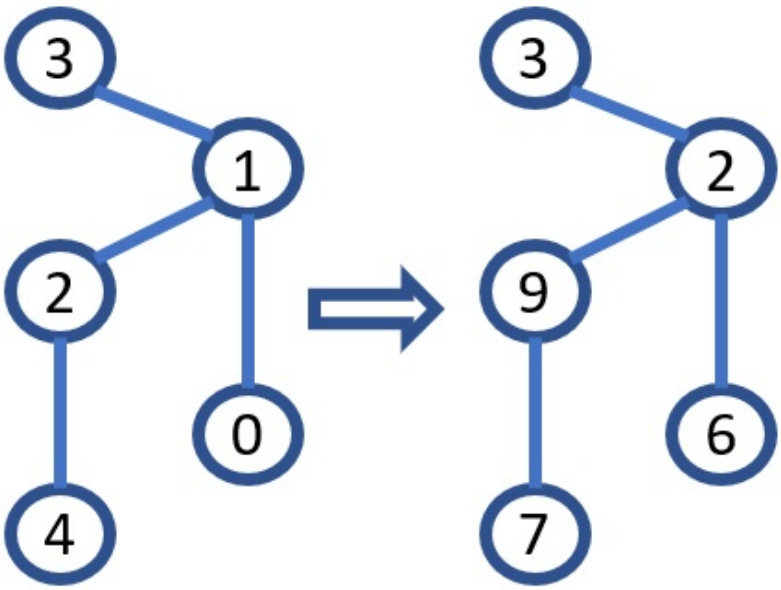
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

Υπάρχουν συνολικά 5 σταθμοί και 4 σύνδεσμοι που ενώνουν τα ζεύγη σταθμών με δείκτες $(0, 1)$, $(1, 2)$, $(1, 3)$ και $(2, 4)$. Κάθε ετικέτα μπορεί να είναι ένας ακέραιος από το 0 μέχρι το $k = 10$.

Για να αναθέσετε τις παρακάτω ετικέτες:

Δείκτης	Ετικέτα
0	6
1	2
2	9
3	3
4	7

η συνάρτηση `label` πρέπει να επιστρέψει $[6, 2, 9, 3, 7]$. Οι αριθμοί στην πιο κάτω εικόνα δείχνουν τους δείκτες (αριστερά) και τις ετικέτες που ανατίθενται σε κάθε σταθμό (δεξιά).



Υποθέστε ότι οι ετικέτες έχουν ανατεθεί όπως παραπάνω και θεωρήστε την ακόλουθη κλήση:

```
find_next_station(9, 6, [2, 7])
```

Αυτό σημαίνει ότι ένα πακέτο βρίσκεται στο σταθμό με ετικέτα 9 και έχει προορισμό το σταθμό με

ετικέτα 6. Οι ετικέτες των σταθμών στη διαδρομή προς τον προορισμό είναι $[9, 2, 6]$. Επομένως, η κλήση θα πρέπει να επιστρέψει 2, δηλαδή την ετικέτα του σταθμού στον οποίο πρέπει να προωθηθεί το πακέτο (ο οποίος έχει δείκτη 1).

Θεωρήστε άλλη μια δυνατή κλήση:

```
find_next_station(2, 3, [3, 6, 9])
```

Η συνάρτηση πρέπει να επιστρέψει 3, καθώς ο σταθμός-προορισμός με ετικέτα 3 είναι γειτονικός του σταθμού με ετικέτα 2 και επομένως θα πρέπει να λάβει το πακέτο απευθείας.

Περιορισμοί

- $1 \leq r \leq 10$

Για κάθε κλήση στη `label`:

- $2 \leq n \leq 1000$
- $k \geq n - 1$
- $0 \leq u[i], v[i] \leq n - 1$ (για κάθε $0 \leq i \leq n - 2$)

Για κάθε κλήση στη `find_next_station`, η είσοδος προέρχεται από μια αυθαίρετα επιλεγμένη προηγούμενη κλήση στην `label`. Θεωρήστε τις ετικέτες που παρήγαγε. Τότε:

- s και t είναι ετικέτες δύο διαφορετικών σταθμών.
- c είναι η ακολουθία των ετικετών όλων των γειτονικών σταθμών του σταθμού με ετικέτα s, σε αύξουσα σειρά.

Για κάθε περίπτωση ελέγχου, το άθροισμα των μηκών όλων των πινάκων c που δίνονται στη συνάρτηση `find_next_station` δεν υπερβαίνει τις $100\ 000$ για όλα τα σενάρια συνολικά.

Υποπροβλήματα

1. (5 βαθμοί) $k = 1000$, κανένας σταθμός δεν έχει πάνω από 2 γείτονες.
2. (8 βαθμοί) $k = 1000$, ο σύνδεσμος i ενώνει τους σταθμούς $i + 1$ και $\left\lfloor \frac{i}{2} \right\rfloor$.
3. (16 βαθμοί) $k = 1\ 000\ 000$, το πολύ ένας σταθμός έχει περισσότερους από 2 γείτονες.
4. (10 βαθμοί) $n \leq 8$, $k = 10^9$
5. (61 βαθμοί) $k = 10^9$

Στο υποπρόβλημα 5 μπορείτε να παρετε μερική βαθμολογία (partial scoring). Έστω m η μέγιστη τιμή ετικέτας που επιστρέφει η `label` σε όλα τα σενάρια. Η βαθμολογία σας σε αυτό το υποπρόβλημα θα υπολογιστεί σύμφωνα με τον ακόλουθο πίνακα:

Μέγιστη ετικέτα	Βαθμολογία
$m \geq 10^9$	0
$2000 \leq m < 10^9$	$50 \cdot \log_{5 \cdot 10^5}(\frac{10^9}{m})$
$1000 < m < 2000$	50
$m \leq 1000$	61

Υποδειγματικός βαθμολογητής

Ο υποδειγματικός βαθμολογητής διαβάζει την είσοδο ως εξής:

- γραμμή 1: r

Ακολουθούν r τμήματα (blocks), που κάθε ένα περιγράφει ένα σενάριο. Η μορφή κάθε τμήματος έχει ως εξής:

- γραμμή 1: n k
- γραμμή $2 + i$ ($0 \leq i \leq n - 2$): $u[i]$ $v[i]$
- γραμμή $1 + n$: q: το πλήθος των κλήσεων στην `find_next_station`.
- γραμμή $2 + n + j$ ($0 \leq j \leq q - 1$): $z[j]$ $y[j]$ $w[j]$: οι **δείκτες** των σταθμών που περιλαμβάνονται στην j-οστή κλήση στην `find_next_station`: ο σταθμός $z[j]$ έχει το πακέτο, ο σταθμός $y[j]$ είναι ο σταθμός-προορισμός του πακέτου και ο σταθμός $w[j]$ είναι ο σταθμός που πρέπει να προωθηθεί.

Η έξοδος του υποδειγματικού βαθμολογητή είναι ως εξής:

- γραμμή 1: m

Ακολουθούν r τμήματα (blocks) που αντιστοιχούν στα διαδοχικά σενάρια της εισόδου. Η μορφή κάθε τμήματος έχει ως εξής:

- γραμμή $1 + j$ ($0 \leq j \leq q - 1$): ο **δείκτης** του σταθμού, του οποίου η **ετικέτα** έχει επιστραφεί με την j-οστή κλήση στην `find_next_station` σε αυτό το σενάριο.

Να σημειωθεί ότι κάθε εκτέλεση του υποδειγματικού βαθμολογητή καλεί και τις δύο συναρτήσεις, `label` και `find_next_station`.