Κλειδιά (keys)

Ο Τιμόθεος ο αρχιτέκτων έχει σχεδιάσει ένα νέο παιχνίδι διαφυγής. Σε αυτό το παιχνίδι υπάρχουν n δωμάτια αριθμημένα από το 0 μέχρι το $n-1$. Αρχικά, κάθε δωμάτιο περιέχει μόνο ένα κλειδί. Κάθε κλειδί έχει έναν τύπο, ο οποίος είναι ένας ακέραιος μεταξύ 0 και $n-1$, συμπεριλαμβανομένων. Ο τύπος του κλειδιού στο δωμάτιο i ($0 \leq i \leq n-1$) είναι $r[i]$. Να σημειωθεί ότι πολλά δωμάτια μπορούν να περιέχουν κλειδιά του ιδίου τύπου, δηλαδή οι τιμές των $r[i]$ δεν είναι υποχρεωτικά μοναδικές.

Υπάρχουν επίσης m **αμφίδρομοι** σύνδεσμοι μέσα στο παιχνίδι, αριθμημένοι από το 0 μέχρι το $m-1$. Ο σύνδεσμος j ($0 \leq j \leq m-1$) συνδέει δύο διαφορετικά δωμάτια $u[j]$ και $v[j]$. Ένα ζευγάρι δωματίων μπορεί να συνδέεται με πολλούς συνδέσμους.

Το παιχνίδι παίζεται από έναν παίκτη που μαζεύει τα κλειδιά και μετακινείται μεταξύ των δωματίων διασχίζοντας τους συνδέσμους. Λέμε ότι ο παίκτης **διασχίζει** τον σύνδεσμο j όταν χρησιμοποιεί αυτόν τον σύνδεσμο για να μετακινηθεί από το δωμάτιο $u[j]$ στο δωμάτιο $v[j]$, ή αντίστροφα. Ο παίκτης μπορεί να διασχίσει τον σύνδεσμο j μόνο αν έχει προηγουμένως μαζέψει ένα κλειδί με τύπο $c[j]$.

Σε οποιοδήποτε σημείο του παιχνιδιού ο παίκτης βρίσκεται σε κάποιο δωμάτιο x και μπορεί να κάνει δύο ενέργειες:

- να μαζέψει το κλειδί από το δωμάτιο x, του οποίου ο τύπος είναι $r[x]$ (εκτός αν το έχει ήδη μαζέψει),
- να διασχίσει έναν σύνδεσμο j, όπου είτε $u[j] = x$ ή $v[j] = x$, αν ο παίκτης έχει ήδη μαζέψει το κλειδί με τύπο $c[j]$. Προσέξτε ότι ο παίκτης **ποτέ** δεν χάνει τα κλειδιά που έχει μαζέψει.

Ο παίκτης **ξεκινά** το παιχνίδι από κάποιο δωμάτιο s χωρίς να έχει κανένα κλειδί στην κατοχή του. Το δωμάτιο t είναι **προσβάσιμο** από το δωμάτιο s, αν ο παίκτης που ξεκινά το παιχνίδι από το δωμάτιο s μπορεί να κάνει μια σειρά από ενέργειες όπως αυτές που περιγράφονται πιο πάνω και να φτάσει στο δωμάτιο t.

Για κάθε δωμάτιο i ($0 \leq i \leq n-1$), έστω $p[i]$ το πλήθος των δωματίων που είναι προσβάσιμα από το δωμάτιο i. Ο Τιμόθεος θέλει να γνωρίζει το σύνολο δεικτών i που επιτυγχάνουν την ελάχιστη δυνατή τιμή $p[i]$ για $0 \leq i \leq n-1$.

Λεπτομέρειες υλοποίησης

Πρέπει να υλοποιήσετε την παρακάτω συνάρτηση:

```
int[] find_reachable(int[] r, int[] u, int[] v, int[] c)
```

- r: ένας πίνακας μήκους n. Για κάθε i ($0 \leq i \leq n-1$), το κλειδί στο δωμάτιο i είναι τύπου $r[i]$.
- u, v: δύο πίνακες μήκους m. Για κάθε j ($0 \leq j \leq m-1$), ο σύνδεσμος j συνδέει τα δωμάτια $u[j]$ και $v[j]$.
- c: ένας πίνακας μήκους m. Για κάθε j ($0 \leq j \leq m-1$), ο τύπος κλειδιού που χρειάζεται για να διασχίσουμε το σύνδεσμο j είναι $c[j]$.
- Αυτή η συνάρτηση πρέπει να επιστρέφει έναν πίνακα s μήκους n. Για κάθε $0 \leq i \leq n-1$, η τιμή του $s[i]$ θα είναι 1 αν για κάθε j τέτοιο ώστε $0 \leq j \leq n-1$ ισχύει $p[i] \leq p[j]$. Αλλιώς, η τιμή του $s[i]$ θα είναι 0.

Παραδείγματα

Παράδειγμα 1

Έστω η παρακάτω κλήση:

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

Αν ο παίκτης ξεκινήσει το παιχνίδι στο δωμάτιο 0, μπορεί να κάνει την ακόλουθη σειρά ενεργειών:

Δωμάτιο όπου βρίσκεται	**Ενέργεια**
0	Μάζεψε το κλειδί τύπου 0
0	Διάσχισε τον σύνδεσμο 0 για το δωμάτιο 1
1	Μάζεψε το κλειδί τύπου 1
1	Διάσχισε τον σύνδεσμο 2 για το δωμάτιο 2
2	Διάσχισε τον σύνδεσμο 2 για το δωμάτιο 1
1	Διάσχισε τον σύνδεσμο 3 για το δωμάτιο 3

Επομένως, το δωμάτιο 3 είναι προσβάσιμο από το δωμάτιο 0. Ομοίως, μπορούμε να δημιουργήσουμε ακολουθίες που δείχνουν ότι όλα τα δωμάτια είναι προσβάσιμα από το δωμάτιο 0, το οποίο σημαίνει ότι $p[0] = 4$. Ο παρακάτω πίνακας δείχνει τα προσβάσιμα δωμάτια για όλα τα δωμάτια εκκίνησης:

Δωμάτιο εκκίνησης i	**Προσβάσιμα δωμάτια**	$p[i]$
0	$[0, 1, 2, 3]$	4
1	$[1, 2]$	2
2	$[1, 2]$	2
3	$[1, 2, 3]$	3

Η ελάχιστη τιμή του $p[i]$ για όλα τα δωμάτια είναι 2, και αυτό επιτυγχάνεται για $i = 1$ ή $i = 2$. Επομένως, η συνάρτηση πρέπει να επιστρέψει $[0, 1, 1, 0]$.

Παράδειγμα 2

```
find_reachable([0, 1, 1, 2, 2, 1, 2],
               [0, 0, 1, 1, 2, 3, 3, 4, 4, 5],
               [1, 2, 2, 3, 3, 4, 5, 5, 6, 6],
               [0, 0, 1, 0, 0, 1, 2, 0, 2, 1])
```

Ο παρακάτω πίνακας δείχνει τα προσβάσιμα δωμάτια:

Δωμάτιο εκκίνησης i	**Προσβάσιμα δωμάτια**	$p[i]$
0	$[0, 1, 2, 3, 4, 5, 6]$	7
1	$[1, 2]$	2
2	$[1, 2]$	2
3	$[3, 4, 5, 6]$	4
4	$[4, 6]$	2
5	$[3, 4, 5, 6]$	4
6	$[4, 6]$	2

Η ελάχιστη τιμή του $p[i]$ για όλα τα δωμάτια είναι 2 και αυτό επιτυγχάνεται για $i \in \{1, 2, 4, 6\}$. Επομένως, η συνάρτηση πρέπει να επιστρέψει $[0, 1, 1, 0, 1, 0, 1]$.

Παράδειγμα 3

```
find_reachable([0, 0, 0], [0], [1], [0])
```

Ο παρακάτω πίνακας δείχνει τα προσβάσιμα δωμάτια:

Δωμάτιο εκκίνησης i	**Προσβάσιμα δωμάτια**	$p[i]$
0	$[0, 1]$	2
1	$[0, 1]$	2
2	$[2]$	1

Η ελάχιστη τιμή του $p[i]$ για όλα τα δωμάτια είναι 1 και αυτό επιτυγχάνεται για $i = 2$. Επομένως, η συνάρτηση πρέπει να επιστρέψει $[0, 0, 1]$.

Περιορισμοί

- $2 \leq n \leq 300\,000$
- $1 \leq m \leq 300\,000$
- $0 \leq r[i] \leq n-1$ για κάθε $0 \leq i \leq n-1$
- $0 \leq u[j], v[j] \leq n-1$ και $u[j] \neq v[j]$ για κάθε $0 \leq j \leq m-1$
- $0 \leq c[j] \leq n-1$ για κάθε $0 \leq j \leq m-1$

Υποπροβλήματα

1. (9 βαθμοί) $c[j] = 0$ για κάθε $0 \leq j \leq m-1$ και $n, m \leq 200$
2. (11 βαθμοί) $n, m \leq 200$
3. (17 βαθμοί) $n, m \leq 2000$
4. (30 βαθμοί) $c[j] \leq 29$ (for all $0 \leq j \leq m-1$) και $r[i] \leq 29$ (για κάθε $0 \leq i \leq n-1$)
5. (33 βαθμοί) Κανένας επιπλέον περιορισμός.

Υποδειγματικός βαθμολογητής

Ο υποδειγματικός βαθμολογητής διαβάζει την εισοδο ως εξής:

- γραμμή 1: $n \; m$
- γραμμή 2: $r[0] \; r[1] \; \ldots \; r[n-1]$
- γραμμή $3+j$ ($0 \leq j \leq m-1$): $u[j] \; v[j] \; c[j]$

Ο υποδειγματικός βαθμολογητής τυπώνει την τιμή που επιστρέφει η συνάρτηση `find_reachable` στην ακόλουθη μορφή:

- γραμμή 1: $a[0] \; a[1] \; \ldots \; a[n-1]$