

Kulcsok(Keys)

Timothy, az építész új szabadulós játékot tervez. A játékban n szoba van, 0-tól $n - 1$ -ig számozva. Kezdetben minden szobában egyetlen kulcs van. Minden kulcs valamilyen típusba tartozik, 0 és $n - 1$ értékek közt, a határokat is beleértve. Az i . szoba ($0 \leq i \leq n - 1$) kulcsának típusa $r[i]$. Több szobában is lehet ugyanolyan típusú kulcs, azaz az $r[i]$ értékek nem feltétlenül különböznek.

Van még m kétirányú ajtó a játékban, 0-tól $m - 1$ -ig számozva. A j . ajtó ($0 \leq j \leq m - 1$) az egymástól különböző $u[j]$ és $v[j]$ szobákat köti össze és $c[j]$ típusú kulccsal használható. Bármely szobapár között több ajtó is lehet.

Az egyszemélyes játékban a játékos gyűjti a kulcsokat és a szobák közt az ajtókon keresztül közlekedik. A játékos a j . ajtón át **közlekedik**, ha az $u[j]$ szobából a $v[j]$ szobába megy, vagy visszamegy rajta. A játékos csak akkor használhatja a j . ajtót, ha már előzőleg összegyűjtött egy $c[j]$ típusú kulcsot.

A játék bármely pontján a játékos valamely x szobában tartózkodik és kétféle műveletet hajthat végre:

- összegyűjtheti az x szobában levő $r[x]$ típusú kulcsot (függetlenül attól, hogy már egyszer ilyent összegyűjtött vagy sem),
- közlekedhet a j . ajtón át, amire $u[j] = x$ vagy $v[j] = x$, ha már előzőleg gyűjtött $c[j]$ típusú kulcsot. A játékos **soha** nem dob el egy, már összegyűjtött kulcsot.

A játékos valamely s szobából **indul**, és kezdetben nincs nála kulcs. A t . szoba **elérhető** az s . szobából, ha a játékos az s szobából indulva a fenti műveleteket használva el tud jutni a t szobába.

Minden egyes i -re ($0 \leq i \leq n - 1$), az i . szobából elérhető szobák számát jelöljük $p[i]$ -vel. Timothy szeretné tudni azon i sorszámokat, melyekre a $p[i]$ értéke minimális ($0 \leq i \leq n - 1$).

Megvalósítás

A következő függvényt kell elkészítened:

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

- r : egy n elemű tömb. Minden i -re ($0 \leq i \leq n - 1$), az i . szobában levő kulcs típusa $r[i]$.
- u, v : két, m elemű tömb. Minden j -re ($0 \leq j \leq m - 1$), a j . ajtó az $u[j]$. és a $v[j]$. szobát köti össze.
- c : egy m elemű tömb. Minden j -re ($0 \leq j \leq m - 1$), a j . ajtó kulcsának típusa $c[j]$.
- A függvény visszatérési értéke az n elemű a tömb. Minden i -re $0 \leq i \leq n - 1$, az $a[i]$ legyen 1, ha minden j -re $p[i] \leq p[j]$ ($0 \leq j \leq n - 1$). Egyébként az $a[i]$ legyen 0.

Példák

1. példa

Tekintsük a következő hívást:

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

Ha a játékos a 0. szobából indul, a következő műveletek sorozatát hajthatja végre:

Aktuális szoba	Művelet
0	Összegyűjti a 0. típusú kulcsot
0	A 0. ajtón keresztül az 1. szobába megy
1	Összegyűjti az 1. típusú kulcsot
1	A 2. ajtón keresztül a 2. szobába megy
2	A 2. ajtón keresztül az 1. szobába megy
1	A 3. ajtón keresztül a 3. szobába megy

Így 3. szoba elérhető a 0. szobából. Hasonlóan készíthetünk útvonalat, ami megmutatja, hogy a 0. szobából minden más szoba elérhető, így $p[0] = 4$. Az alábbi táblázat tartalmazza az egyes kezdőszobákból elérhető szobákat:

Az i szobából indulva	Elérhető szobák	$p[i]$
0	[0, 1, 2, 3]	4
1	[1, 2]	2
2	[1, 2]	2
3	[1, 2, 3]	3

A $p[i]$ legkisebb értéke az összes szobát tekintve 2, és ezt az $i = 1$ illetve az $i = 2$ helyen veszi fel. Így a függvény visszatérési értéke [0, 1, 1, 0].

2. példa

```
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])
```

Az alábbi táblázat tartalmazza az egyes kezdőszobákból elérhető szobákat:

Az i szobából indulva	Elérhető szobák	$p[i]$
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

A $p[i]$ legkisebb értéke az összes szobát tekintve 2, és ezt az $i \in \{1, 2, 4, 6\}$ helyeken veszi fel. Így a függvény visszatérési értéke [0, 1, 1, 0, 1, 0, 1].

3. példa

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

Az alábbi táblázat tartalmazza az elérhető szobákat:

Az i szobából indulva	Elérhető szobák	$p[i]$
0	[0, 1]	2
1	[0, 1]	2
2	[2]	1

A $p[i]$ legkisebb értéke az összes szobát tekintve 1, és ezt az $i = 2$ helyen veszi fel. Így a függvény visszatérési értéke [0, 0, 1].

Korlátok

- $2 \leq n \leq 300\,000$
- $1 \leq m \leq 300\,000$
- $0 \leq r[i] \leq n - 1$ ($0 \leq i \leq n - 1$)
- $0 \leq u[j], v[j] \leq n - 1$ és $u[j] \neq v[j]$ ($0 \leq j \leq m - 1$)
- $0 \leq c[j] \leq n - 1$ ($0 \leq j \leq m - 1$)

Részfeladatok

1. (9 pont) $c[j] = 0$ ($0 \leq j \leq m - 1$ és $n, m \leq 200$)
2. (11 pont) $n, m \leq 200$

3. (17 pont) $n, m \leq 2000$

4. (30 pont) $c[j] \leq 29$ ($0 \leq j \leq m - 1$) és $r[i] \leq 29$ ($0 \leq i \leq n - 1$)

5. (33 pont) Nincs további megkötés.

Mintaértékelő

A mintaértékelő az alábbi formában olvassa a bemenetet:

- Az 1. sor: n m
- A 2. sor: $r[0]$ $r[1]$ \dots $r[n - 1]$
- A $3 + j$. sor ($0 \leq j \leq m - 1$): $u[j]$ $v[j]$ $c[j]$

A mintaértékelő a következő formában írja ki a `find_reachable` visszatérési értékeit:

- 1. sor: $s[0]$ $s[1]$ \dots $s[n - 1]$