

Raktai

Architektas Timotis sukūrė naują pabėgimo žaidimą. Šiame žaidime yra n kambarių, sunumeruotų nuo 0 iki $n - 1$. Iš pradžių kiekviename kambaryje yra lygiai vienas raktas. Kiekvienas raktas yra tam tikros rūšies, aprašomos sveikuoju skaičiumi nuo 0 iki $n - 1$ imtinai. i -ajame ($0 \leq i \leq n - 1$) kambaryje esančio rakto rūšis yra $r[i]$. Atkreipkite dėmesį, kad keli kambariai gali turėti tos pačios rūšies raktus, t.y. $r[i]$ reikšmės nebūtinai skiriasi.

Taip pat žaidime yra m **dvikrypčių** koridorių, sunumeruotų nuo 0 iki $m - 1$. j -asis ($0 \leq j \leq m - 1$) koridorius jungia du skirtingus kambarius – $u[j]$ -ąjį ir $v[j]$ -ąjį. Tuos pačius du kambarius gali jungti daugiau nei vienas koridorius.

Žaidime dalyvauja vienintelis žaidėjas, kuris renka raktus ir juda tarp kambarių eidamas koridoriais. Sakome, kad žaidėjas **pereina** j -ąjį koridorių, kai jis pasinaudoja šiuo koridoriumi, kad patektų iš $u[j]$ -ojo kambario į $v[j]$ -ąjį kambarį arba atvirkščiai. Žaidėjas gali pereiti j -ąjį koridorių tik jei jis jau turi paėmęs rūšies $c[j]$ raktą.

Bet kuriuo žaidimo momentu žaidėjas yra kažkuriame x -ajame kambaryje ir gali atlikti dviejų rūšių veiksmus:

- paimti x -ajame kambaryje esantį raktą, kurio rūšis yra $r[x]$ (nebent jis šį raktą jau yra paėmęs),
- pereiti j -uoju koridoriumi, jei $u[j] = x$ arba $v[j] = x$ ir žaidėjas jau yra paėmęs rūšies $c[j]$ raktą. Atkreipkite dėmesį, kad žaidėjas **niekada** nepraranda rakto, kurį jau turi paėmęs.

Žaidėjas **pradedą** žaidimą kuriame nors kambaryje s , neturėdamas nei vieno rakto. t -asis kambarys yra **pasiekiamas** iš s -ojo kambario, jei žaidėjas, pradedantis žaidimą s -ajame kambaryje, gali atlikti seką aukščiau aprašytų veiksmų ir pasiekti t -ąjį kambarį.

Kiekvienam i -ajam kambariui ($0 \leq i \leq n - 1$), pažymėkime skaičių kambarių, pasiekiamų iš i -ojo kambario, simboliu $p[i]$. Timotis norėtų sužinoti, su kuriais indeksais i yra įgyjama minimali $p[i]$ reikšmė, kai $0 \leq i \leq n - 1$.

Realizacija

Parašykite šią procedūrą:

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

- r : n ilgio masyvas. Kiekvienam i ($0 \leq i \leq n - 1$), i -ajame kambaryje esančio rakto rūšis yra $r[i]$.
- u, v : du m ilgio masyvai. Kiekvienam j ($0 \leq j \leq m - 1$), j -asis koridorius jungia $u[j]$ -ąjį ir $v[j]$ -ąjį kambarius.

- c : m ilgio masyvas. Kiekvienam j ($0 \leq j \leq m - 1$), rakto rūšis, reikalinga norint pereiti j -ąjį koridorių, yra $c[j]$.
- Ši procedūra turi grąžinti n ilgio masyvą a . Kiekvienam $0 \leq i \leq n - 1$, $a[i]$ reikšmė turi būti 1, jei kiekvienam j , tenkinančiam $0 \leq j \leq n - 1$, galioja $p[i] \leq p[j]$. Priešingu atveju, $a[i]$ reikšmė turi būti 0.

Pavyzdžiai

Pavyzdys nr. 1

Panagrinėkime tokį iškvietimą:

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

Jei žaidimas pradedamas 0-iniame kambaryje, žaidėjas gali atlikti šiuos veiksmus:

Dabartinis kambarys	Veiksmas
0	Paimti rūšies 0 rakta
0	Pereiti 0-iniu koridoriumi į 1-ąjį kambarį
1	Paimti rūšies 1 rakta
1	Pereiti 2-uuju koridoriumi į 2-ąjį kambarį
2	Pereiti 2-uuju koridoriumi į 1-ąjį kambarį
1	Pereiti 3-uuju koridoriumi į 3-ąjį kambarį

Todėl 3-iasis kambarys yra pasiekiamas iš 0-inio kambario. Panašiais būdais galime sukonstruoti veiksmų sekas, kuriomis visi kambariai yra pasiekiami iš 0-inio, todėl $p[0] = 4$. Žemiau pateikta lentelė rodo pasiekiamų kambarių sąrašus kiekvienam startiniam kambariui:

Startinio kambario nr. i	Pasiekiami kambariai	$p[i]$
0	[0, 1, 2, 3]	4
1	[1, 2]	2
2	[1, 2]	2
3	[1, 2, 3]	3

Mažiausia $p[i]$ reikšmė tarp visų kambarių yra 2, ir ji yra gaunama, kai $i = 1$ arba $i = 2$. Taigi, funkcija turi grąžinti [0, 1, 1, 0].

Pavyzdys 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])
```

Žemiau pateikta lentelė rodo pasiekiamus kambarius:

Startinio kambario nr. i	Pasiekiami kambariai	$p[i]$
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

Mažiausia $p[i]$ reikšmė tarp visų kambarių yra 2, ir ji yra gaunama, kai $i \in \{1, 2, 4, 6\}$. Taigi, funkcija turi grąžinti [0, 1, 1, 0, 1, 0, 1].

Pavyzdys 3

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

Žemiau pateikta lentelė rodo pasiekiamus kambarius:

Startinio kambario nr. i	Pasiekiami kambariai	$p[i]$
0	[0, 1]	2
1	[0, 1]	2
2	[2]	1

Mažiausia $p[i]$ reikšmė tarp visų kambarių yra 1, ir ji yra gaunama, kai $i = 2$. Taigi, funkcija turi grąžinti [0, 0, 1].

Ribojimai

- $2 \leq n \leq 300\,000$
- $1 \leq m \leq 300\,000$
- $0 \leq r[i] \leq n - 1$ visiems $0 \leq i \leq n - 1$
- $0 \leq u[j], v[j] \leq n - 1$ ir $u[j] \neq v[j]$ visiems $0 \leq j \leq m - 1$

- $0 \leq c[j] \leq n - 1$ visiems $0 \leq j \leq m - 1$

Dalinės užduotys

1. (9 taškai) $c[j] = 0$ visiems $0 \leq j \leq m - 1$ ir $n, m \leq 200$
2. (11 taškų) $n, m \leq 200$
3. (17 taškų) $n, m \leq 2000$
4. (30 taškų) $c[j] \leq 29$ (visiems $0 \leq j \leq m - 1$) ir $r[i] \leq 29$ (visiems $0 \leq i \leq n - 1$)
5. (33 taškai) Jokių papildomų ribojimų.

Pavyzdinė vertinimo programa

Pavyzdinė vertinimo programa skaito įvestį tokiu formatu:

- 1-oji eilutė: $n \ m$
- 2-oji eilutė: $r[0] \ r[1] \ \dots \ r[n - 1]$
- $(3 + j)$ -oji ($0 \leq j \leq m - 1$) eilutė: $u[j] \ v[j] \ c[j]$

Pavyzdinė vertinimo programa išveda `find_reachable` grąžintą reikšmę tokiu formatu:

- 1-oji eilutė: $a[0] \ a[1] \ \dots \ a[n - 1]$