



## Simurga

Senovinė persų legendą iš Šanameho sako, kad persų legendinis didvyris Zalas yra beprotiškai įsimylėjęs Kabilo princesę Rudabą. Kai Zalas paprašė jos rankos, jos tėvas davė jam užduotį.

Persijoje yra  $n$  miestų, sunumeruotų nuo 0 iki  $n - 1$ , ir  $m$  dvikrypčių kelių, sunumeruotų nuo 0 iki  $m - 1$ . Kiekvienas kelias jungia du skirtingus miestus. Kiekviena miestų pora yra sujungta daugiausiai vienu keliu. Kai kurie keliai yra *karališki keliai*, kuriais keliauja karaliaus šeima. Zalas turi nustatyti, kurie keliai yra karališki.

Zalas turi žemėlapi su visais miestais ir keliais Persijoje. Jis nežino, kurie keliai yra karališki, bet jam gali pagelbėti Simurga, geroji mitinė paukštė, Zalo globėja. Tačiau Simurga nenori iš karto atskleisti visos karališkų kelių aibės. Vietoj to, ji pasako, kad visų karališkų kelių aibė yra *auksinė aibė*. Kelių aibė yra auksinė tada ir tik tada, jei:

- ją sudaro lygiai  $n - 1$  kelias, ir
- kiekvienai miestų porai įmanoma pasiekti vieną miestą iš kito keliaujant tik šios aibės keliais.

Be to, Zalas gali klausti Simurgos. Kiekvieną kartą klausdamas:

1. Zalas pasirenka vieną galimą *auksinių* kelių aibę, ir tada
2. Simurga pasako Zalui kiek kelių pasirinktoje auksinėje aibėje yra karališki.

Padėkite Zalui surasti karališkų kelių aibę užduodami Simurgai ne daugiau  $q$  klausimų. Vertintojas atliks Simurgos vaidmenį.

## Realizacija

Parašykite šią procedūrą:

```
int[] find_roads(int n, int[] u, int[] v)
```

- $n$ : miestų skaičius,
- $u$  ir  $v$ :  $m$  ilgio masyvai. Visiems  $0 \leq i \leq m - 1$ ,  $u[i]$  ir  $v[i]$  yra miestai, sujungti keliu  $i$ .
- Ši procedūra turi grąžinti  $n - 1$  ilgio masyvą, sudarytą iš karališkų kelių numerių, pateiktų bet kokia tvarka.

Sprendimas gali atlikti daugiausiai  $q$  užklausų šiai vertintojo procedūrai:

```
int count_common_roads(int[] r)
```

- $r$ :  $n - 1$  ilgio masyvas, sudarytas iš auksinės aibės kelių numerių (bet kokia tvarka).
- Ši procedūra gražina karališkų kelių skaičių aibėje  $r$ .

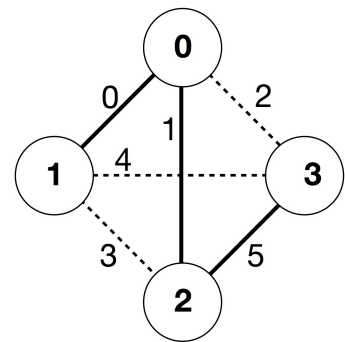
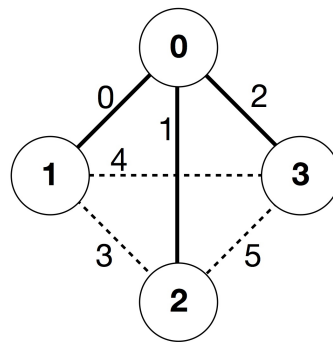
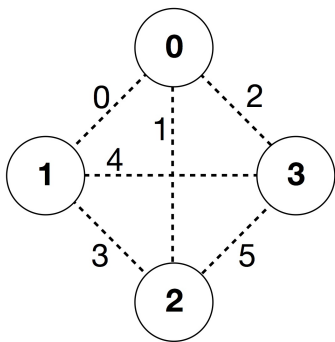
## Pavyzdys

```
find_roads(4, [0, 0, 0, 1, 1, 2], [1, 2, 3, 2, 3, 3])
```

`find_roads(...)`

`count_common_roads([0, 1, 2]) = 2`

`count_common_roads([5, 1, 0]) = 3`



Šiame pavyzdyje yra 4 miestai ir 6 keliai. Pažymėkime  $(a, b)$  kelią, jungiantį miestus  $a$  ir  $b$ . Keliai sunumeruoti nuo 0 iki 5 šia tvarka:  $(0, 1)$ ,  $(0, 2)$ ,  $(0, 3)$ ,  $(1, 2)$ ,  $(1, 3)$  ir  $(2, 3)$ . Kiekviena auksinė aibė turi  $n - 1 = 3$  kelius.

Darykime prielaidą, kad karališki keliai turi numerius 0, 1 ir 5, t.y. keliai  $(0, 1)$ ,  $(0, 2)$  ir  $(2, 3)$ . Tada:

- `count_common_roads([0, 1, 2])` gražina 2. Ši užklausa yra apie kelius 0, 1 ir 2, t.y. kelius  $(0, 1)$ ,  $(0, 2)$  ir  $(0, 3)$ . Du iš šių kelių yra karališki.
- `count_common_roads([5, 1, 0])` gražina 3. Ši užklausa yra apie visus karališkus kelius.

Procedūra `find_roads` turi gražinti  $[5, 1, 0]$  arba bet kokį 3 ilgio masyvą, sudarytą iš šių trijų elementų.

Atkreipkite dėmesį, kad šios užklauskos yra neleidžiamos:

- `count_common_roads([0, 1])`: čia  $r$  ilgis nėra 3.
- `count_common_roads([0, 1, 3])`: čia  $r$  nepapršo auksinės aibės, nes neįmanoma nuvykti iš miesto 0 į 3 tik keliais  $(0, 1)$ ,  $(0, 2)$ ,  $(1, 2)$ .

## Ribojimai

- $2 \leq n \leq 500$
- $n - 1 \leq m \leq n(n - 1)/2$
- $0 \leq u[i], v[i] \leq n - 1$  (kiekvienam  $0 \leq i \leq m - 1$ )
- Visiems  $0 \leq i \leq m - 1$ , kelias  $i$  jungia du skirtingus miestus (t.y.  $u[i] \neq v[i]$ ).
- Tarp dviejų miestų yra daugiausiai vienas kelias.
- Keliais galima nukeliauti iš bet kurio vieno į bet kurį kitą miestą.

- Karališkų kelių aibė yra auksinė aibė.
- `find_roads` turi iškviešti `count_common_roads` ne daugiau kaip  $q$  kartų. Kiekvienoje užklausoje  $r$  aprašoma kelių aibė turi būti auksinė.

## Subtasks

1. (13 taškų)  $n \leq 7$ ,  $q = 30\,000$
2. (17 taškų)  $n \leq 50$ ,  $q = 30\,000$
3. (21 taškas)  $n \leq 240$ ,  $q = 30\,000$
4. (19 taškų)  $q = 12\,000$  ir tarp kiekvienos miestų poros yra kelias
5. (30 taškų)  $q = 8000$

## Pavyzdinis vertintojas

Pavyzdinis vertintojas skaito duomenis tokiu formatu:

- 1-a eilutė:  $n$   $m$
- $i + 2$ -a eilutė (kiekvienam  $0 \leq i \leq m - 1$ ):  $u[i]$   $v[i]$
- $m + 2$ -a eilutė:  $s[0]$   $s[1]$   $\dots$   $s[n - 2]$

Čia  $s[0], s[1], \dots, s[n - 2]$  yra karališkų kelių numeriai.

Pavyzdinis vertintojas išveda YES, jei `find_roads` iškviečia `count_common_roads` ne daugiau nei 30 000 kartų ir gražina teisingą karališkų kelių aibę. Kitais atvejais jis išveda NO.

Atkreipkite dėmesį, kad pavyzdinio vertintojo procedūra `count_common_roads` netikrina, ar  $r$  turi visas auksinės aibės savybes. Vietoj to ji suskaičiuoja ir išveda karališkų kelių numerių skaičių masyve  $r$ . Tačiau jei priduta programa iškviečia `count_common_roads` su numerių aibe, kuri nenusako auksinės aibės, vertintojas pateiks 'Wrong Answer' (neteisingas atsakymas).

## Techninė pastaba

Procedūra `count_common_roads` C++ ir Pascal kalbose efektyvumo dėlei naudoja *iškvietimo pagal nuorodą* metodą. Šią procedūrą vis tiek galite iškviešti įprastu būdu. Yra garantuojama, kad vertintojas nepakeis  $r$  reikšmės.